

KYMENLAAKSON AMMATTIKORKEAKOULU

Tekniikan ylempi AMK-tutkinto / Teknologiaosaamisen johtaminen

Jonna Lahtinen

JÄSENREKISTERISOVELLUKSEN TOTEUTUS OHJELMISTOPROJEKTINA

Opinnäytetyö 2012

TIIVISTELMÄ

KYMENLAAKSON AMMATTIKORKEAKOULU

Teknologiaosaamisen johtaminen

LAHTINEN, JONNA

Opinnäytetyö

Työn ohjaaja

Toimeksiantaja

Marraskuu 2012

Avainsanat

Jäsenrekisterisovelluksen toteutus ohjelmistoprojektina

93 sivua

Yliopettaja Simo Ollila

Kotkan Melojat ry

ohjelmistoprojekti, jäsenrekisteri, tietoturva

Opinnäytetyön aiheena oli luoda uusi jäsenrekisteri Kotkan Melojat ry:lle. Päättötyö on kirjoitettu projektihallinnan ja tietoturvan näkökulmasta. Kotkan Melojat ry on yhdistys, jonka toiminta perustuu vapaaehtoisuuteen ja talkoohenkeen, joten projektina työ poikkesi jonkin verran kaupallisista ohjelmistoprojekteista erityisesti aikataulun, budjetin ja muiden resurssien osalta. Työ seuraa normaaleja projektinhallinnan askeleita esitutkimuksesta ylläpitoon.

Kotkan Melojien alkuperäinen jäsenrekisteri oli Excel-pohjainen taulukko, jonka ylläpito oli hankalaa. Jokaisella yhdistyksen hallituksen jäsenellä oli oma versionsa taulukosta, eivätkä niihin tehdyt muutokset näkyneet muiden taulukoissa. Jäsenrekisteri sisälsi paljon vanhentunutta tietoa. Uusi jäsenrekisteri on web-pohjainen sovellus, joka käyttää MySQL -tietokantaa. Näin kaikilla on käytössään sama versio rekisteristä ja tietojen ylläpito on helpompaa ja muutokset näkyvät kaikille samanaikaisesti.

Uudessa jäsenrekisterissä on mahdollisuus ylläpitää jäsentietoja ja tehdä yhdistyksen laskutus. Sovelluksen avulla voidaan luoda myös erilaisia tilastoja, laskutusraportteja ja lähettää jäsenkirjeitä jäsenten sähköpostiin tai tulostaa niitä postitusta varten. Jäsenille voidaan luoda omia käyttäjätunnuksia, joiden avulla jäsenet voivat itse tarkastaa omat tietonsa ja päivittää niitä. Sovelluksessa on myös oma foorumi jäsenille, jonka kautta jäsenet voivat olla yhteydessä toisiinsa ja keskustella eri aiheista.

ABSTRACT

KYMENLAAKSON AMMATTIKORKEAKOULU

University of Applied Sciences

Master of Engineering Programme in Technology Administration

LAHTINEN, JONNA

Creating a Membership Register as a Software Project

Master Thesis

93 pages

Supervisor

Simo Ollila, Principal Lecturer

Commissioned by

Kotkan Melojat ry

November 2012

Keywords

software project, membership register, data security

The purpose of this master's thesis, written from the perspective of project management and data security, was to create a new web-based membership register for the paddling club Kotkan Melojat ry. As the activities of the club are founded on voluntary and community spirit, the project management differed in some respects from commercial projects particularly in terms of the timetables, budget and personnel resources. This thesis covers the normal project management steps from basic planning to maintenance.

The original membership register was an Excel table of which there were several different versions. As a result, keeping the register up-to-date presented a particular challenge. A further problem was that the old membership register included a lot of incorrect and outdated information. To address these challenges, the new membership management software was developed as a web-based system which uses a MySQL database. The new system facilitates the updating of personal information and makes the changes implemented available to all board members immediately.

The new membership register also includes an electric member application, membership management and invoicing software. It provides the opportunity to create various statistics, generate billing reports and write letters to members. The members can use their own usernames and passwords to access the software and update their own personal information. Besides this, they also have a forum where they can chat with each other.

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

1	JOHDANTO	12
2	PROJEKTIN MÄÄRITELMÄ JA PROJEKTITYYPIT	13
3	PROJEKTIN OSITUS JA VAIHEISTUS	14
3.1	Vaiheittainen ositus	15
3.2	Järjestelmiin osittaminen	16
3.3	Rakenteellinen osittaminen	16
3.4	Työlajin mukainen ositus	17
3.5	Ohjelmistoprojektin ositus	17
3.5.1	Vesiputousmalli (waterfall model)	17
3.5.2	Evo-malli (evolutionary delivery)	19
3.5.3	Prototyyppimalli	19
3.6	Jäsenrekisteriprojektin ositus	21
4	ESITUTKIMUS	22
4.1	Nykytilanteen kuvaaminen	22
4.1.1	Alkuperäinen jäsenrekisteri	22
4.1.2	Kotkan Melojat ry:n kotisivut	23
4.1.3	Kotkan Melojat ry: jäseneksi hyväksyminen	24
4.1.4	Eroaminen yhdistyksestä	24
4.1.5	Laskutus	24
4.1.6	Seuran jäsenten välinen yhteydenpito	25
4.1.7	Sidosryhmät	25
4.1.8	Yhteenvetoa ongelmista ja projektin tavoitteista	26
4.2	Lait, määräykset ja standardit	27
4.2.1	Yhdistyslaki (503/1989)	28
4.2.2	Henkilötietolaki (523/1999)	28
4.2.3	Rekisterinpitäjän velvollisuudet	28

4.2.4	Rekisteröidyn oikeudet	29
4.2.5	Rekisteri- ja tietosuojaseloste	30
4.2.6	Hyvä tiedonhallintatapa	32
4.2.7	Tilisiirto	33
4.2.8	Viitenumero	34
4.2.9	Sähköpostilasku	34
5	MÄÄRITTELY- JA SUUNNITTELUVAIHE	35
5.1	Toiminnallinen määrittely	36
5.2	Tunnuksettomat käyttäjät	37
5.3	Jäsentason käyttäjät	40
5.3.1	Omien tietojen muokkaus	40
5.3.2	Salasanan muuttaminen: jäsentason näkymä	41
5.3.3	Palautteen lähetys	42
5.3.4	Keskustelupalsta, jäsentason näkymä	42
5.4	Ylläpitotason käyttäjät	45
5.4.1	Käyttäjälistaus	45
5.4.2	Käyttäjätunnusten luonti, muokkaus ja poisto	46
5.4.3	Jäsenlistaus	47
5.4.4	Jäsentietojen lisäys, muokkaus ja poisto	49
5.4.5	Palautelaatikko: ylläpitotason näkymä	51
5.4.6	Palautteeseen vastaaminen	51
5.4.7	Keskustelupalstan hallinta	52
5.4.8	Jäsentietojen haku	53
5.4.9	Laskutus	54
5.4.10	Laskutuksen teko ja maksetut maksut	55
5.4.11	Jäsenkirjeet	58
5.5	Ei-toiminnalliset vaatimukset	59
5.5.1	Laatuvaatimukset	59
5.5.2	Käytettävyys	60
5.5.3	Helppokäyttöisyys, loogisuus ja opittavuus	61
5.5.4	Esteettömyys	61
5.5.5	Virhetilanteet	61
5.5.6	Suorituskyky ja vasteajat	62

5.5.7	Muita ei-toiminnallisia vaatimuksia	62
5.6	Tietoturva	63
5.6.1	Identiteettivarkaudet	64
5.6.2	Tietovuodot ja tietomurrot	64
5.6.3	Salasanojen turvallisuus	64
5.6.4	Istuntokaappaukset	66
5.7	Tietokannan suunnittelu ja normalisointi	67
5.8	Tekninen määrittely	69
5.8.1	Webhotelli	69
5.8.2	Notepad++	70
5.8.3	PHP	71
5.8.4	JavaScript	71
5.8.5	MySQL	71
5.8.6	CSS	71
5.8.7	Merkistökoodaus	71
5.9	Testaussuunnitelma	72
5.9.1	Lasilaatikkotestaus	74
5.9.2	Funktionaalinen testaus eli mustalaatikkotestaus	74
5.9.3	Web-selaimet palvelun testauksessa	75
5.10	Riskienhallinta ohjelmistoprojekteissa	76
5.10.1	Aikatauluriskit	78
5.10.2	Riittämättömät resurssit	78
5.10.3	Integraatiot ja tekniset alustat	79
5.10.4	Muutoksen hallinnan riskit	80
5.10.5	Riskit puutteellisesta dokumentoinnista	81
5.10.6	Riskien välttäminen jatkokehityksessä	82
6	TOTEUTUSVAIHE	82
6.1	Tietokannan luonti ja migratointi	82
6.2	Ohjelmointi ja ohjelmoinnin aikainen testaus	83
6.3	Tilisiirtolomakkeen toteutus	84
7	TESTAUSVAIHE	84
7.1	Jäsenrekisterisovelluksen funktionaalinen testaus ja betatestaus	85

7.2	Html-koodin validointi ja selaintestaus	85
8	KÄYTTÖÖNOTTOVAIHE	86
9	YLLÄPITO JA JATKKEHITYS	86
9.1	Varmuuskopiot	86
9.2	Jatkokehityskohteita	87
9.3	Tietoturvan uudelleenarviointi	88
10	MUITA NÄKÖKULMIA	88
11	JOHTOPÄÄTÖKSET JA YHTEENVETO	89

TERMIT JA LYHENTEET

Acid-testi	Testi, jolla testataan käytössä olevia ja tulevia www-standard.de .
ASCII	Tietokoneiden käyttämä merkistö, joka sisältää aakkosia, numeroita ja joitakin erikoismerkkejä. Nykyisin käytössä olevat merkistöt sisältävät kaikki ovat ASCII:n 128 ensimmäistä merkkiä sekä lisäksi laajennuksia niihin.
BIC-koodi (Bank Identifier Code)	Pankin yksilöivä 8 tai 11 merkkiä pitkä koodi.
Brute-force -menetelmä	Salasanojen purkuun käytetty menetelmä, jonka prosessorien laskentatehot mahdollistavat. Menetelmällä yksinkertaisesti kokeillaan kaikkia mahdollisia kirjain, numero ja merkkiyhdistelmiä oikean salasanan löytämiseksi.
CERT-FI	Viestintävirastossa toimiva kansallinen tietoturvaviranomainen, joka mm. valvoo ja tiedottaa tietoturvaloukkauksista.
CSS (Cascading Style Sheets)	Tyylitiedosto, jonka avulla määritellään ulkoasu merkkauksieliselle rakenteelle.
CSV (Comma-separated values)	Tiedostomuoto yksinkertaisen taulukkomuotoisen tiedon tallentamiseksi tekstitiedostoon.
Entiteetti	Erikoismerkkejä voidaan kirjoittaa nk. merkikielien eli entiteettien avulla. Jotkut merkit eivät sisälly koodattaessa käytettävään merkistöön tai niille on varattu erityismerkitys koodikielessä, joten ne korvataan entiteet-

timerkillä jonka avulla selain osaa tulkita oikean merkin.

Evästeet (cookies)

Selaimen palvelimelle lähettämää tietoa, joka tallennetaan käyttäjän koneelle.

HTML (Hyper Text Markup Language)

Web- sivujen tekoon tarkoitettu merkkäuskieli.

IBAN (International Bank Account Number)

Kansainvälinen pankkitilinumero, joka koostuu tilinpitäjäpankin maakoodista, kaksimerkkisestä tarkisteesta ja tilinumerosta.

Kansainvälinen tilinumero, jota käytetään rajat ylittävissä maksuissa EU- ja ETA-alueella.

ISO-200022

XML-pohjainen sanomavälitysjärjestelmä, joka on tarkoitettu yritysten ja pankkien väliseen viestintään.

ISO-5589-1

Eräs merkistökoodausstandardi.

ISO-5589-15

ISO-5589-1 -merkistökoodauksen seuraajaksi kehitetty standardi, joka sisältää joitakin lisämerkkejä ISO-5589-1 -standardiin.

Istunto (session)

Käytäntö, jolla luodaan pysyvä yhteys palvelimen ja käyttäjän välille.

JavaScript

Web-sivuilla käytetty komentosarjakieli.

Migratointi	Migratointi tarkoittaa mm. tietojen siirto esim. ohjelman tai tietokannan eri versioiden välillä siten, että ne ovat edelleen käytettävissä muodossa.
Normalisointi	Tietokannan normalisointimalli, jonka avulla selkeytetään tietokannan rakennetta.
OECD (Organisation for Economic Cooperation and Development)	Taloudellisen yhteistyön ja kehityksen järjestö, joka perustettiin kehittämään jäsenmaidensa talouskasvua ja vapaakauppaa ja lisäämään yhteiskunnallista hyvinvointia. Suomi on ollut OECD:n jäsenmaana vuodesta 1969 asti.
PHP	The PHP Groupin kehittämä palvelinohjain ohjelmointikieli dynaamisten web-sivujen tekoon.
Rainbow table hyökkäys	salasanoihin kohdistuva murtomenetelmä, jossa tiivistä algoritmeilla tiivistetyistä salasanoista on etukäteen laskettu tarkistussummat rainbow table -tietokantaan, josta niitä verrataan esim. web-palvelusta saatuihin tarkistussummiin.
Sanakirjahyökkäys (dictionary attack)	salasanojen murtotekniikka, jonka pohjana käytetään tuhansia sanoja sisältävää sanalista. Sanalistan sanoja kokeillaan yksi kerrallaan, alusta loppuun ja lopusta alkuun, isoilla ja pienillä kirjaimilla jne. oikean salasan löytämiseksi.

SQL	IBM:n alun perin kehittämä kyselykieli, jonka avulla voidaan hallita tietokantoja ja tehdä niihin kyselyjä.
URL-osoite	Yksilöllinen osoite jollekin internetissä sijaitsevalle tiedostolle.
UTF-8	Eräs merkistökoodausstandardi.
Validaattori	Validaattori on ohjelma, jonka avulla voidaan tarkastaa, että web-sivun koodauksessa on käytetty standardin mukaista kieltä.
WaSP (The Web Standard Project)	Organisaatio, joka toteuttaa mm. web-selaimille tarkoitettuja Acid -testejä.
W3C World Wide Web Consortium	Yhteisö, joka kehittää ja määrittelee yhteisiä ja yhteensopivia sääntöjä ja teknologioita web-sivuja ja palveluita varten.

1 JOHDANTO

Kotkan melojat ry on vuonna 1934 perustettu yhdistys, jonka toiminta on talkoohenkistä ja perustuu vapaaehtoisuuteen. Yhdistykseen kuuluu noin sata jäsentä ja se järjestää melonnan alkeiskursseja, koskikursseja ja melontaretkiä jäsenilleen. Yhdistyksellä on Kotkassa kaksi kajakkivajaa, joissa säilytetään seuran kalustoa sekä jäsenten omia kajakkeja säilytysmaksua vastaan. Yhdistyksen omistuksessa on koski-, poolo- ja merikajakkeja, joita jäsenet voivat käyttää vapaasti maksettuaan jäsen- ja kalustomaksut. Talvisin järjestetään uimahalliharjoituksia, joissa harjoitellaan pelastautumista ja eskimokäännöksiä. Seuran jäseneksi pääseminen edellyttää näyttöä sitoutumisesta seuran toimintaan, joka tarkoittaa osallistumista seuran järjestämille kursseille, viikkomelontoihin tai muihin tapahtumiin. Kotkan Melojat ry on jäsenenä Suomen melonta- ja soutuliitto ry:ssä, joka toimii soutu- ja melontaseurojen kattojärjestönä Suomessa.

Kotkan Melojat ry:n jäsenrekisterinä on toiminut Excel-pohjainen taulukko, jonka tiedot eivät olleet pysyneet ajan tasalla taulukon hankalan päivityksen vuoksi. Jäsentietojen ylläpito kuuluu Kotkan Melojat ry:n hallitukselle ja jäsenrekisterinä toimivasta Excel-tilusta oli olemassa useita versioita. Jos yksi hallituksen jäsen teki muutoksia taulukkoon, ne eivät näkyneet muiden taulukoissa. Jäsenillä ei ollut pääsyä tarkastelemaan omia tietojaan, eikä yhteystietojen muutoksista ym. muistettu ilmoittaa jäsenrekisteristä vastaaville. Jäsenten laskutus hoidettiin samalla Excel-tilustalla, siihen tehtyjen makrojen avulla. Laskutuksen teko oli hankalaa ja vei paljon aikaa, sillä laskut oli luotava yksitellen.

Olin työttömänä, kun aloitin opintoni teknologiaosaamisen johtamisen koulutusohjelmassa, joten päättötyöaiheen löytäminen ei ollut helppoa. Samoihin aikoihin, kun aloitin opintoni, minut valittiin Kotkan Melojat ry:n sihteeriksi ja tutustuin yhdistyksen toimintaan sen hallituksen näkökulmasta. Sihteerinä tehtäviini kuuluu juuri jäsenrekisterin ylläpito ja laskutus. Niinpä päättötyöni aihe syntyi kuin itsestään.

Tämän päättötyön tarkoituksena oli luoda Kotkan Melojat ry:lle yksi yhtenäinen web-pohjainen jäsenrekisteri, jonka avulla voidaan helposti hallinnoida yhdistyksen laskutusta ja jäsentietoja. Jäsenrekisteriin oli tarkoitus luoda myös keskustelualue, jossa yhdistyksen jäsenet voivat keskustella melonnasta, varusteista, retkeilystä ja muusta aiheeseen liittyvästä.

Tässä päättötyössä pohditaan jäsenrekisterisovelluksen tekoa erityisesti ohjelmistoprojektin toteutuksen ja tietoturvan näkökulmasta. Koska yhdistyksen toiminta on talkoohenkistä ja vapaaehtoisuuteen perustuvaa se poikkeaa jonkin verran kaupallisiin tarkoituksiin tehtävistä projekteista, eikä projektinhallinnassa ole kiinnitetty huomiota taloudellisiin näkökohtiin tai riskeihin. Myöskään resursointiin liittyvät asiat eivät olleet oleellisia projektin toteutuksessa. Päättötyössäni käsittelen jäsenrekisterisovelluksen luontia projektina aina suunnittelusta käyttöönottoon ja jatkokehitykseen saakka.

2 PROJEKTIN MÄÄRITELMÄ JA PROJEKTITYYPIT

Projekti määritellään kertaluonteiseksi työksi, joka tehdään jokin ainutlaatuisen tuloksen aikaansaamiseksi. Projektilla on aina alku ja loppu (Suomen Projekti-Instituutti Oy 2012). Projektit voidaan jakaa erilaisiin projektityyppeihin niiden luonteen perusteella. Tyypillisimpiä projektityyppejä ovat:

- Kehitysprojektit, joiden avulla kehitetään organisaation toimintaa tai luodaan kokonaan jokin uusi valmistukseen ja myyntiin tarkoitettu tuote
- Tutkimusprojekti, joka voi edeltää tuotekehitysprojektia ja jonka tarkoituksena on löytää jokin uusi ratkaisu tai arvioida edellytyksiä esimerkiksi taloudelliseen toimintaan.
- Toimitusprojekti, jolla tarkoitetaan tuotteen toimitusta asiakkaalle projektin muodossa.

Pohjonen on pohtinut perusteita ja tarpeita, jotka johtavat tietojärjestelmän kehitykseen:

Tietojärjestelmän kehittäminen alkaa tarpeesta kehittää uutta tai ylläpitää vanhaa. Perusteet tälle tarpeelle voivat olla moninaiset: kehitystyö voi käynnistyä esimerkiksi asiakkaan tarpeista, uusien teknisten mahdollisuuksien myötä, kehittämispaineiden vuoksi, jonkin toisen kehityshankkeen yhteydessä esiin tulleiden tarpeiden perusteella tai jonkin laajemman kartoitustyön, kuten esimerkiksi kokonaistutkimuksen perusteella. (Pohjonen 2007: 26).

Jäsenrekisterin toteutus pohjaa todelliseen tarpeeseen helpottaa jäsenrekisterin ylläpitoa ja laskutusta. Samalla tarkoituksena oli muokata toimintaan liittyviä prosesseja uutta järjestelmää tukeviksi ja tarjota seuran jäsenille uuden yhteydenpitotavan kes-

kustelupalstan avulla. Näin ollen sovelluksen toteutus on tyypillinen kehitysprojekti, jonka tarkoituksena on uudistaa, kehittää ja tehostaa yhdistyksen toimintaa. Projektin perusteet ovat samalla moninaiset: asiakkaan tarve, uudet tekniset mahdollisuudet sekä kehittämispaineet.

3 PROJEKTIN OSITUS JA VAIHEISTUS

Projektin toteutusta voidaan selkeyttää projektiosituksella, jonka avulla projekti voidaan jakaa erilaisiin kokonaisuuksiin, jotka muodostavat hierarkkisen järjestyksen. Ositus helpottaa projektin kaikkien osa-alueiden hallintaa. Projektihallinnan käsikirja kuvailee projektin ositusta seuraavasti:

Projektiositusta voidaan käyttää projektin kaikkien osapuolien suunnitelmien, aikataulujen, budjettien ja raporttien pohjana. Näin se on projektihallinnan keskeinen informaatioväline. Mitään yhtä ainoaa oikeata tapaa osittaa tietty projekti ei ole, vaan ositus voidaan tehdä usealla eri tavalla. (Pelin 2011: 92).

Projektiositukselle voidaan asettaa useita eri tavoitteita ja sen tarkoituksena on toimia myös pohjana kaikelle projektissa tapahtuvalle toiminnalle. Pelin (2011: 91-92) asettaa projektin ositukselle seuraavia tavoitteita:

- Projektiositus vaiheistaa projektin. Kussakin projektin vaiheessa voi ositus olla erilainen.
- Projektiositus jakaa projektin organisatorisesti selviin vastuukokonaisuuksiin ja osaprojekteihin.
- Projektiositus jakaa projektin aikataulut osa-aikatauluiksi, joihin on merkitty niiden keskinäiset riippuvuussuhteet.
- Projektiositus luo puitteet kustannusohjaukselle määrittelemällä seuraavat kustannuskohteet (ns. työpaketit)
- Projektiositus antaa projektin työlle hierarkkisen jäsentelyn ja koodauksen (WBS-no)
- Projektiosituksen avulla voidaan integroida ajallinen ja taloudellinen suunnittelu ja ohjaus.

Projektin ositukseen voidaan käyttää seuraavien perusmenetelmien yhdistelmiä, jotka eivät ole toisensa poissulkevia:

- vaiheittainen ositus
- järjestelmiin osittaminen
- rakenteellinen ositus
- työlajin mukainen ositus (Pelin 2011: 92).

Suomen Projekti-Instituutti Oy:n mukaan projektin elinkaaresta voidaan tunnistaa projektista toiseen toistuvia päävaiheita.

Tyypilliset päävaiheet projektinjohtamisen tasolla ovat aloitus, suunnittelu, toteutus ja lopetus. Toteutustasolla projekti voi jakautua yksityiskohtaisempiin vaiheisiin projektityypin ja valitun toteutusmallin mukaisesti. (Suomen Projekti-Instituutti Oy 2012).

3.1 Vaiheittainen ositus

Projektin vaiheittainen ositus tehdään jakamalla projektin toiminnot aikajärjestyksessä eteneviin vaiheisiin. Näin saadaan katettua projektin koko elinkaari. Pelin kuvaa vaiheittaista ositusta seuraavasti:

Vaiheittaisella osituksella tarkoitetaan projektin jakamista peräkkäisiin vaiheisiin. Näin projektiin tulee ajallisesti rajatut, itsenäiset osat. Tavallisia vaiheita ovat mm. esitutkimus, suunnittelu, toteutus ja käyttöönotto. (Pelin 2011: 93).

Ohjelmistotuotannossa projektin vaiheittainen ositus on yleensä selkeää, koska projektiin liittyvät tehtävät ja osaprojektit muodostavat usein lähes luontaisesti ajallisen hierarkian. Eri työvaiheita ei useinkaan voida suorittaa, ennen kuin edellinen on saatu valmiiksi tai ainakin tiettyyn pisteeseen asti. Ohjelmistoprojekteissa eri vaiheet seuraavat toisiaan tai ovat osittain päällekkäisiä. Toisaalta ohjelmistoprojekteissa on tärkeää, että edelliseen vaiheeseen voidaan palata tarvittaessa uudelleen. Pohjosen mukaan vaihejako selkeyttää erityisesti tietojärjestelmien kehitystä:

Tietojärjestelmien elinkaaren vaihejaolla pyritään ensisijaisesti määrittämään järjestelmän kehittämisen tehtävät, niiden ajoitus ja riippuvuudet toisistaan. Vaihejako luo myös mielekkään viitekehyksen kuhunkin vaiheeseen liittyvien käsitteiden, ongelmien, menetelmien ja työkalujen hallintaan (Pohjonen 2007: 26).

3.2 Järjestelmiin osittaminen

Projektin osittaminen järjestelmiin tarkoittaa sen pilkkomista erillisiin systeemeihin. Pelin (2001: 93) kuvaa järjestelmiin ositusta seuraavasti:

Järjestelmiin osittamisella projekti eritellään systeemeittään. Systeemi on toiminnallisesti itsenäinen kokonaisuus, joka tyypillisesti liittyy poikittaisesti projektin eri rakenteisiin. Esimerkiksi tiedonsiirtojärjestelmä, lämmitysjärjestelmä, jäähdytysjärjestelmä, valvonta- ja ohjausjärjestelmä jne.

Ohjelmisto- ja järjestelmäprojektit ovat luonnostaan monimutkaisia, joten niiden hallinta ja seuranta voi olla hankalaa. Useista tällaisista projekteista voidaan kuitenkin erotella projektin osat juuri järjestelmittään. Esimerkiksi yksi sovellus voi sisältää osia eri toimialojen tarpeisiin, kuten taloushallinto, asiakaspalvelu, myynti ja varastonhallinta. Sovelluksen voidaan usein osittaa myös järjestelmittään erilaisten käyttäjien mukaan: verkkokaupan asiakkaat tarvitsevat erilaiset toiminnot ja käyttöliittymät kuin siitä vastaavat myyjät.

3.3 Rakenteellinen osittaminen

Rakenteellinen ositus on projektin ositusta fyysisiin osiin. Pelin (2011: 93) kuvaa rakenteellista ositusta seuraavasti:

Laajassa projektissa voidaan ensimmäiseksi eritellä maantieteellisesti erillään sijaitsevat osat kuten eri rakennukset. Sitten eritellään rakennusten osat, konekokonaisuudet jne. Nämä pilkotaan edelleen osiinsa seuraavalla WBS-tasolla. Rakenteellinen ositus on aina perustana projektin osittamisessa.

3.4 Työlajin mukainen ositus

Projektit voidaan osittaa myös työlajin mukaisesti. Työlajeihin perustuva ositus ohjelmisto ja järjestelmäprojekteissa on luontaista, sillä niihin liittyy usein useita toisistaan hyvinkin poikkeavia tehtäviä. Eri työlajeja ovat esimerkiksi projektinhallinto, suunnittelu, ohjelmointi, testaus, koulutus, markkinointi, tiedostus jne.

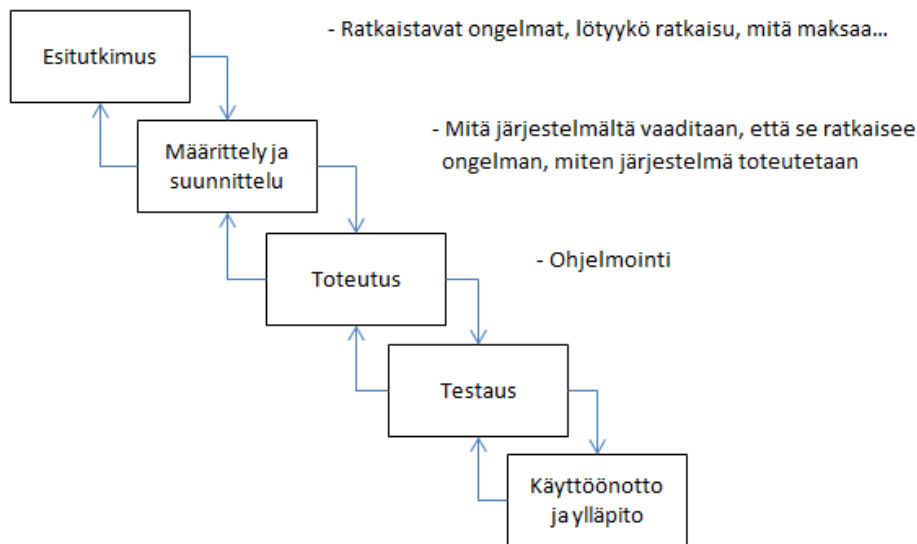
3.5 Ohjelmistoprojektin ositus

Pohjosen mukaan (2007: 39) tietojärjestelmien elinkaari on pitkään nähty kokonaisvaltaisena mallintamiskohteenä, johon on otettu oppia mm. talonrakennuksesta. Pohjosen muistuttaa myös, että malleissa toiminnot on järjestetty selkeiksi kokonaisuuksiksi, mutta kyseessä on kuitenkin vain malli, eikä se aina välttämättä sovellu käytäntöön sellaisenaan.

Ohjelmistotuotannon projekteihin on olemassa lukuisia valmiita periaatemalleja, joita käytetään pohjana projektien ositukseen. Perinteisimpiä malleja ovat muun muassa: vesiputous-, evo- ja prototyypimallit, joita on kuvattu tarkemmin seuraavissa kappaleissa.

3.5.1 Vesiputousmalli (waterfall model)

Tunnetuin ohjelmistotuotannon vaihejakomalli on niin kutsuttu vesiputousmalli. Vesiputousmallista on olemassa lukuisia muunnelmia, mutta kaikista löytyvät yleensä ainakin seuraavat osat: määrittely-, suunnittelu- ja toteutusvaihe. Alla olevassa kuvassa (Kuva 1) on mukailtu versio perinteisestä vesiputousmallista, jota käytetään yleisesti kuvaamana ohjelmistoprojektin vaiheita vaiheittaisen osituksen periaatteita noudattaen:



Kuva 1 Mukailtu kuva alun perin Winston W. Roycen 1970-luvulla kehittämästä vesiputousmallista.

Vesiputousmallissa eri vaiheet seuraavat toisiaan. Malli mahdollistaa myös palaamisen edelliseen vaiheeseen, mikäli ohjelmistokehitys sitä vaatii. Ohjelmistoprojekti aloitetaan esitutkimuksella, jonka tarkoituksena on kartoittaa lähtötilanne ja asiakasvaatimukset. Esitutkimuksen tuloksena saatetaan saada myös tietoa onko jonkin projektin aloittaminen taloudellisesti järkevää. Esitutkimus voi toisaalta olla oma projektinsa, joka on tyypiltään tutkimusprojekti.

Esitutkimuksen jälkeen seuraa kehitettävän sovelluksen tarkempi määrittelyvaihe. Määrittelyvaiheessa määritellään tarkemmin projektin hallintaan liittyvät seikat, kuten aikataulut, resurssit, budjetit ja riskinhallinta. Ohjelmistoprojekteissa määrittelyvaiheeseen liittyvät myös teknisten ja toiminnallisten vaatimusten määrittely.

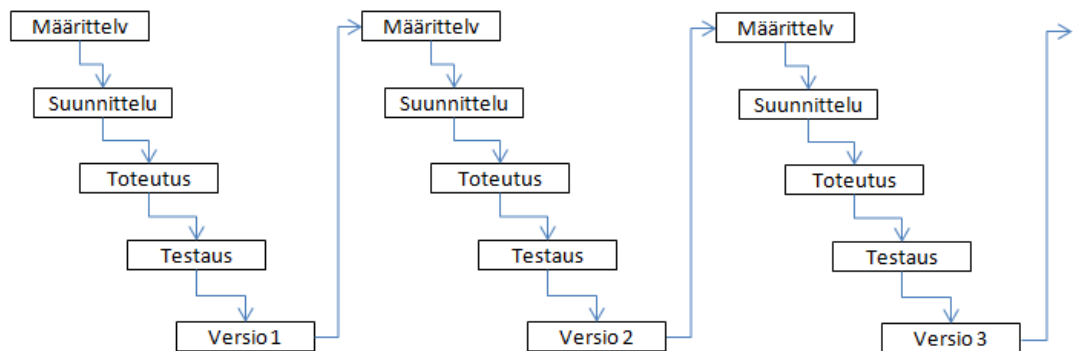
Toteutusvaiheessa toteutetaan itse tuote. Toteutusvaiheen kanssa tehdään usein osittain samanaikaisesti myös testausta. Kun testaus on saatu loppuun, aloitetaan käyttöönottovaihe, eli julkistetaan valmis tuote ja siirrytään sen ylläpitoon.

Malli ei vastaa ohjelmistoprojektin toteutusta kovinkaan käytännönläheisesti, sillä se ei ota kantaa mm. eri vaiheiden päällekkäisyyksiin ja sen käyttö on muutenkin hankalaa sen joustamattomuuden vuoksi, mutta silti se on yhä edelleen yksi eniten käytetyistä malleista ohjelmistotuotannon projekteissa. Vesiputousmallia on kritisoitu myös siitä, että asiakkaan kannalta konkreettisia tuloksia saadaan vasta melko myöhäisessä

vaiheessa ja näin ollen suuri osa projektista tehdään asiakkaalta piilossa eikä asiakas pääse osallistumaan toimintaan riittävästi.

3.5.2 Evo-malli (evolutionary delivery)

Kuten vesiputousmallistakin, myös evo-mallista on olemassa useita erilaisia versioita. Evo-mallia voitaisiinkin pitää yhtenä vesiputousmallin versiona, sillä se muodostuu useista peräkkäisistä vesiputouksista (kts. 3.5.1 Vesiputousmalli (waterfall model) s. 17).

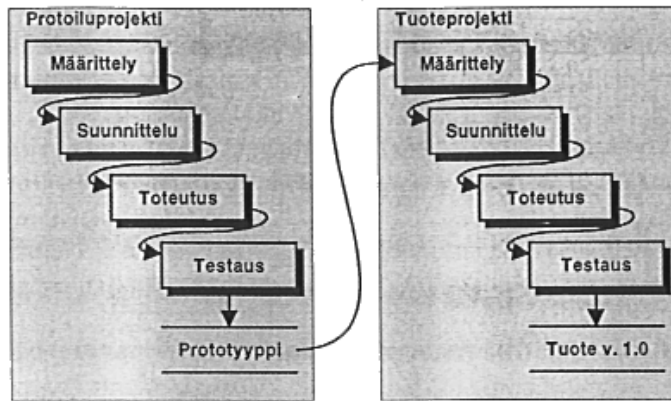


Kuva 2 Mukailtu kuva evo-mallista.

Jokaisen vesiputouksen lopputuloksena syntyy uusi versio sovelluksesta tai järjestelmästä. Toisin sanoen edellisestä versiosta saatu palaute aloittaa uuden version kehityksen. Evo-mallin periaate on esitetty yllä olevassa kuvassa (Kuva 2). Käytännössä kaikki ohjelmistotuotannon tuotekehitysprojektit toteutetaan evo-mallin mukaisesti. Evo-mallin käytössä on samat ongelmat kuin sen pohjana olevassa vesiputousmallistakin.

3.5.3 Prototyyppimalli

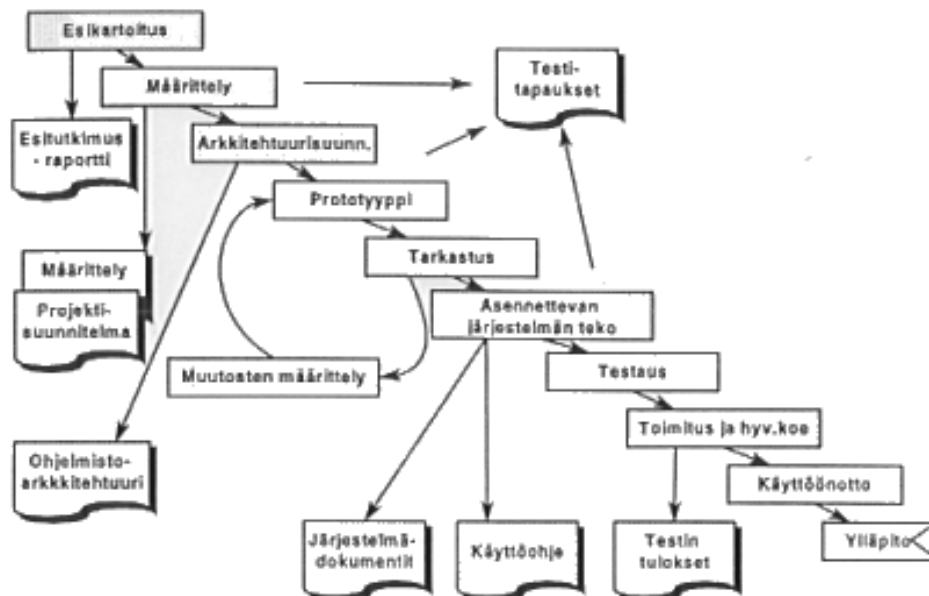
Prototyyppimallissa tuotteesta kehitetään yksinkertaistettu versio, jonka avulla pyritään havainnollistamaan tuotteen toimintaa ja ominaisuuksia esimerkiksi loppukäyttäjille. Prototyyppiä kehitetään edelleen, eli luodaan uusi prototyyppi saadun palautteen perusteella ja kehittelyä jatketaan, kunnes tulokseksi on saatu tavoitteiden mukainen järjestelmä. Prototyyppimallin toiminta on esitetty alla olevassa kuvassa (Kuva 3).



Kuva 3 Mukailtu kuva prototyyppimallista (Haikala & Märijärvi 1997: 30).

Prototyyppimallia käytetään erityisesti uusien järjestelmien kehityksessä. Mallin toiminnasta johtuen sitä voidaan käyttää kuitenkin lähinnä vain pienten ja yksinkertaisten järjestelmien rakentamiseen. Prototyyppimallia käytetään usein myös epäselvien asiakasvaatimusten määrittelemiseksi sekä käyttöliittymäsuunnittelun tukena. Yksinkertaisimmiksi prototyypeiksi voitaisiin kutsua esimerkiksi sovelluksen rautalanka- tai näyttökuvia, joiden avulla asiakkaille esitellään ja mallinnetaan lopullisen tuotteen toimintaa.

Alla olevassa kuvassa (Kuva 4) on prototyyppimallin tarkempi kuvaus, jossa prototyyppi kehitetään lopulliseksi tuotteeksi.



Kuva 4 Prototyyppimallin tarkempi kuvaus (Haikala & Märijärvi 1997: 31).

Prototyypin mallin ongelmana on usein, että projekti jää pyörimään ”prototyyppi – tarkastus – määrittely” -silmukkaan, eli tuotetta määritellään ja parannellaan jatkuvasti lisää ja lisää, eikä toteutusvaiheeseen päästä ollenkaan. Loputtoman määrittelyn kehä voidaan kuitenkin Haikalan ja Märijärven (1997: 30) mukaan estää:

Ongelma on kuitenkin ratkaistavissa sopimusteknisesti: sopimuksessa varataan kiinteä määrä työviikkoja prototyypin kehittämiseen. Jos asiakas vielä tämänkin jälkeen haluaa jatkaa prototyypin kehittämistä, siirrytään tunti-laskutukseen.

Toinen prototyypin mallin ongelma on se, että järjestelmän ”kaksinkertainen” kehittäminen vie aikaa ja resursseja, eikä ns. rautalankamalli aina välttämättä paljasta kehitettävän järjestelmän todellisia ongelmia.

3.6 Jäsenrekisteriprojektin ositus

Jäsenrekisteriprojektin ositukseen käytettiin erilaisten perusmenetelmien yhdistelmiä. Projektin vaiheittainen ositus noudattaa pitkälti vesiputousmallin (kts. 3.5.1 Vesiputousmalli (waterfall model) s. 17) periaatteita. Projektista oli tunnistettavissa selkeästi seuraavat vaiheet:

- Esitutkimus
- Määrittely- ja suunnitteluvaihe
- Toteutusvaihe
- Testausvaihe
- Käyttöönotto
- Ylläpito ja jatkokehitys

Projektissa havaittiin myös erilaisia ositusvaihtoehtoja mm. sidosryhmien mukaan (esim. jäsenet, hallituksen jäsenet jne.) sekä toiminnallisten kokonaisuuksien mukaan (esim. jäsentiedot, laskutus, keskustelupalsta jne.).

Tässä opinnäytetyössä on pohdittu projektin toteutusta vaiheittaisen osituksen näkökulmasta, koska se noudattelee pitkälti myös projektin ositusta eri työtehtävien mukaan (esim. esitutkimus, suunnittelu, ohjelmointi, testaus, tiedotus, käyttöönotto jne.) ja on siksi luonnollinen valinta. Tulevaisuudessa voisi kuvitella, että jäsenrekisterin

jatkokehitys tulee noudattamaan evo-mallin mukaista mallia, jossa edellisestä versioista saatu palaute aloittaa uuden version kehityksen.

4 ESITUTKIMUS

Esitutkimusvaiheen tarkoituksena oli kartoittaa jäsenrekisterin nykytila ja ongelmat, joita uuden järjestelmän avulla pyritään ratkomaan. Samalla suunniteltiin yleisellä tasolla järjestelmän yleisiä vaatimuksia ja niiden toteuttamismahdollisuuksia. Esitutkimusvaihe liittyi tiiviinä osana määrittelyvaiheeseen erityisesti eri sidosryhmien vaatimusten selvityksen osalta. Seuraavissa kappaleissa on kuvattu tarkemmin esitutkimusvaiheen tuloksia.

4.1 Nykytilanteen kuvaaminen

Esitutkimus aloitettiin selvittämällä lähtötilannetta, eli olemassa olevan jäsenrekisterin tilaaja yhdistyksen toimintatapoja. Nykytilan kuvaamisen avulla saatiin tietoa olemassa olevista ongelmista, joihin toivottiin ratkaisuja uudelta jäsenrekisterisovellukselta.

4.1.1 Alkuperäinen jäsenrekisteri

Kuten aiemmin on mainittu (kts. 1 JOHDANTO s. 12) Kotkan Melojat ry:n alkuperäisenä jäsenrekisterinä toimi Excel-pohjainen taulukko, jonka tiedot eivät olleet ajantasaisia ja taulukosta oli olemassa useita eri versioita. Jäsenillä ei ollut mahdollisuutta päivittää omia tietojaan rekisteriin, eikä esimerkiksi osoitetietojen muutoksesta muistettu ilmoittaa hallituksen jäsenille. Laskutus hoidettiin myös saman taulukon avulla, mikä oli hankalaa, koska taulukkoon luodut makrot eivät tukeneet massalaskutusta.

Alkuperäisestä jäsenrekisteristä löytyi 115 jäsenen tiedot, mutta vain 97:lle oli merkitty yhteystiedot, joista suuri osa niistäkin oli vanhentuneita. Jäsenluettelosta löytyi useita jo kauan sitten seurasta eronneita jäseniä ja seuraan parina edellisenä vuonna liittyneitä ei ollut siinä ollenkaan.

4.1.2 Kotkan Melojat ry:n kotisivut

Kotkan Melojat ry:llä oli olemassa kotisivut (Kuva 5), joilta löytyy tietoa mm. seuran tapahtumista ja yhteyshenkilöistä. Sivuilta löytyy myös toimintakertomus ja tietoa jäsenyydestä ja jäsenmaksuista.



Kuva 5 Kotkan Melojat ry:n kotisivut.

Kotisivut on tehty HTML-kielellä ja ne käyttävät tyylisivuinaan CSS-tiedostoa. Jäsenrekisterin suunnitteluvaiheessa mietittiin, voitasiko samaa tyyli-tiedostoa ja ulkoasua käyttää pienin poikkeuksin myös jäsenrekisterissä, mutta koska sovelluksen ja enemmän ”markkinoinnillisempiin” tarkoituksiin tarkoitettujen kotisivujen käyttötarkoitukset eroavat toisistaan merkittävästi, päätettiin jäsenrekisterille tehdä oma tyyli-tiedosto ja graafinen ilme, joka selkeyttää jäsenrekisterin toimintoja. Jäsenrekisterisovelluksen käytettävyyden kannalta kuvat ja värit olisivat vaikeuttaneet sovelluksen käyttöä ja tehneet siitä sekavan.

Kotkan Melojat ry:n kotisivut on tarkoitus säilyttää ennallaan, sillä niitä voivat käyttää myös ulkopuoliset seuran toiminnasta kiinnostuneet ihmiset ilman tunnuksia ja salasanoja. Kotisivuilta näkee myös helpommin seuran tapahtumakalenterin, ilman kirjautumista jäsenrekisteriin. Kaikilla jäsenillä ei myöskään ole tunnuksia jäsenrekisteriin. Integraatio olemassa olevien kotisivujen ja jäsenrekisterin välillä on toteutettu pelkän linkityksen avulla siten, että kotisivuilta on linkki jäsenrekisterin kirjautumissivulla ja linkki uuteen jäsenhakemukseen.

4.1.3 Kotkan Melojat ry: jäseneksi hyväksyminen

Kotkan Melojat ry:n jäseneksi pääseminen edellyttää osallistumista seuran järjestämään toimintaan ja jäseneksi haetaan erillisellä jäsenhakemuksella. Jäseneksi hyväksymisestä päättää yhdistyksen hallitus yhdistyslain mukaisesti (Finlex 1989: L 503 3:12). Seuran jäsenhakemus oli yhdistyksen kotisivuilta löytyvä PDF-tiedosto, joka lähetettiin täytettynä sähköpostitse tai tulostettuna postitse jäsenrekisterin ylläpitäjälle. Hakemuksen tiedot kirjoitettiin uudelleen Excel-pohjaiseen jäsenrekisteriin. Tietojen uudelleenkirjoitus lisäsi työn ja virheiden mahdollisuuksien määrää. Koska uusia hakemuksia tulee yleensä, erityisesti alkeiskurssin jälkeen, useita yhdellä kertaa ja pienen ajan sisällä, ne saattoivat hukkua muun sähköpostin sekaan ja jäädä huomaamatta.

4.1.4 Eroaminen yhdistyksestä

Jos jäsen haluaa erota Kotkan Melojat ry:stä, on eroanomus tehtävä kirjallisesti ja lähetettävä yhdistyksen hallitukselle yhdistyslain mukaisesti (Finlex 1989: L 3:13). Seuran alkuperäisessä jäsenrekisterissä oli useiden, seurasta jo vuosia aiemmin eronneiden, jäsenten tietoja. Eronneille jäsenille lähetettiin yhä jäsenmaksulaskuja, koska heidän tietonsa olivat yhä rekisterissä. Henkilötietolain (Finlex 1999: L 523 2:9-10, 7:34) mukaan rekisterinpitäjän kannalta tarpeettomien henkilötietojen säilytys on kielletty ja ne on hävitettävä.

4.1.5 Laskutus

Kotkan Melojat ry:n jäseniltä laskutetaan jäsenmaksu ja kalustomaksu niiltä jäseniltä, jotka haluavat käyttää seuran kalustoa. Lisäksi seura laskuttaa säilytysmaksua jäsenten omien kalustojen säilytyksestä seuran vajoilla. Yhdistyksen laskutus on aiemmin hoi-

dettu jäsenrekisterinä toimivaan Excel-taulukkoon tehdyn makron avulla, jolla laskut oli luotava yksi kerrallaan.

4.1.6 Seuran jäsenten välinen yhteydenpito

Seuran jäsenet tapaavat toisiaan lähinnä satunnaisesti tai kesäisin kerran viikossa järjestettävien viikkomelontojen yhteydessä. Jäsenten toivomuksena oli, että saataisiin yhteinen foorumi, jossa jäsenet voisivat jakaa käyttökokemuksia retkeily- ja melontavarusteita, suunnitella melontaretkiä, myydä varusteita tai etsiä melontaseuraa. Sähköpostitse tapahtuva viestintä ei toiminut, sillä viesti menee vain siihen merkityille vastaanottajille, eivätkä viestit aina tavoittaneet kaikkia tarvitsevia.

4.1.7 Sidosryhmät

Osana esitutkimusvaihetta oli eri sidos- ja viiteryhmien tunnistaminen ja niiden tarpeiden kartoittaminen ja kuvaaminen. Sidosryhmiksi määriteltiin seuraavat ryhmät ja niiden erityistarpeet:

- Hallituksen jäsenet
 - o ylläpitää jäsenrekisterin tietoja
 - o hoitaa laskutusta
 - o lähettää jäsenkirjeitä
 - o saada tilastotietoja seuran jäsenistä (esim. jäsenten lukumäärä, alkeiskurssilaisten määrä)
- Jäsenet
 - o tarkastaa ja ylläpitää omia yhteystietoja
 - o yhteydenpito muiden jäsenten kanssa
- Seuran toiminnasta kiinnostuneet ulkopuoliset (potentiaaliset jäsenet)
 - o saada tietoa seuran jäsenyydestä
 - o saada tietoa seuran toiminnasta

Sidosryhmien tarkempia tarpeita ja toiveita kartoitettiin kyselyillä ja keskusteluilla.

Monimutkaisemmista toiminnoista luotiin periaatteellisia rautalankamalleja (vrt. 3.5.3 Prototyypimalli s. 19), joiden avulla tarpeita tarkennettiin lisää.

4.1.8 Yhteenvetoa ongelmista ja projektin tavoitteista

Esitutkimusvaiheesta saatiin arvokasta tietoa ongelmista, joihin toivottiin ratkaisua uudelta jäsenrekisterisovellukselta. Ongelmat asettivat projektille selkeämmät tavoitteet ja toimivat hyvänä lähtökohtana sovelluksen tarkemmalle määrittelylle. Alla on listattuna esitutkimuksesta ilmi käyneitä ongelmia sekä pohdintoja ongelmien ratkaisemiseksi.

- **Ongelma:** Jäsenrekisterin tiedot eivät pysy ajan tasalla ja rekisteristä oli olemassa useita versioita. Hallituksen jäsenet tekevät osoitteenmuutokset ym. omiin taulukoihinsa, eivätkä tiedot mene kaikkien hallituksen jäsenten tietoon. Jäsenrekisterissä on lisäksi useita seurasta jo kauan sitten eronneiden jäsenten tietoja. Jäsenillä ei ole mahdollisuutta nähdä omia tietojaan, saati päivittää niitä.

Ratkaisu: Uusi jäsenrekisteri on web-pohjainen sovellus, johon luodaan erilaisia käyttäjätasoja ja niiden mukaan määriteltyjä oikeuksia jäsenrekisterin toimintoihin. Jäsenrekisterin tiedot tallennetaan tietokantaan, jolloin kaikilla on käytössään sama versio jäsenrekisteristä ja muutokset näkyvät kaikille yhtä aikaa.

Kotkan Melojien hallituksen jäsenille luodaan tunnukset, joilla on pääsy jäsenrekisterin kaikkiin tietoihin ja oikeudet lisätä, muuttaa ja poistaa niitä. Jäsenille tehdään omat tunnukset, joiden avulla jokainen seuran jäsen voi tarkistaa omat tietonsa ja muuttaa niitä tarvittaessa. Uuden jäsenrekisterin myötä jäsentietojen käsittely ja ylläpito helpottuu ja tiedot pysyvät paremmin ajan tasalla. Eronneiden jäsenten tiedot ja käyttäjätunnukset poistetaan rekisteristä heti, kun eroanomus on käsitelty.

- **Ongelma:** Laskutus on hankalaa, sillä Excel -taulukoon tehdyt makrot eivät tue massalaskutusta. Laskuista ei jää kopiota mihinkään, eikä niistä saada minkäänlaista laskuraporttia, jonka avulla voitaisiin seurata laskujen maksamista.

Ratkaisu: Uuteen jäsenrekisteriin päätettiin tehdä laskutusjärjestelmä massalaskujen luontia ja tulostusta varten. Lisäksi sovelluksesta voidaan tulostaa erillisiä laskuraportteja ja merkitä maksetut laskut jäsenten tietoihin.

- **Ongelma:** Jäsenhakemukset tulevat sihteerin sähköpostiin PDF-tiedostoina, joista tiedot on kirjoitettava uudelleen jäsenrekisteriin, josta aiheutuu turhaa päällekkäistä työtä. Lisäksi hakemuksia tulee kerrallaan useita ja ne hukkuvat muun sähköpostin sekaan.

Ratkaisu: Uuteen jäsenrekisteriin luodaan sähköinen jäsenhakemuslomake, jonka tiedot menevät suoraan jäsenrekisterin tietokantaan. Uudet hakemukset näkyvät jäsenrekisterin hallintapuolella statuksella ”Uusi hakemus”, joten uusiin hakemuksiin voidaan reagoida hallituksen puolesta helpommin. Kun hakija on virallisesti hyväksytty yhdistyksen jäseneksi, status muutetaan.

- **Ongelma:** Yhdistyksen jäsenten välinen yhteydenpito oli hankalaa, sillä se tapahtui lähinnä sähköpostitse tai puhelinrunkien avulla, eikä tieto kulje toivotusti kaikille. Monet jäsenet eivät uskalla esimerkiksi turvallisuussyistä lähteä yksin merelle melomaan, joten melontaretket rajoittuvat viikkomelontoihin, jos muiden melojien yhteystietoja ei ole.

Yhdistyksen hallitukselle lähetetään tiedotteita muilta seuroilta tai yhdistyksiltä, mutta niiden saattaminen seuran jäsenten tietoon on hankalaa.

Ratkaisu: Uuden jäsenrekisterin yhteyteen luodaan keskustelupalsta, jossa jäsenet voivat keskustella keskenään eri aiheista kuten varusteista tai lähettää myynti- tai ostoilmoituksia, etsiä melontaseuraa yms. Ylläpito voi lisätä sivustolle yleisiä ilmoituksia kursseista, urheilutapahtumista ym. joita yhdistyksen hallitukselle lähetetään sähköpostitse muista melontaan, ulkoiluun ja liikuntaan keskittyneistä seuroista ja organisaatioista.

4.2 Lait, määräykset ja standardit

Lähtötilanteen ja ongelmien lisäksi esitutkimusvaiheessa selvitettiin jäsenrekisteriin vaikuttavat lait, säädökset ja muut ohjeistukset, joita jäsenrekisterin teossa ja käytössä on noudatettava.

Jäsenrekisterin käyttöön ja ylläpitoon sekä yhdistyksen toimintaan liittyvät olennaisesti yhdistyslaki (Finlex 1989: L 503) sekä henkilötietolaki (Finlex 1999: L 523), joiden vaikutusta pohditaan seuraavissa kappaleissa. Lisäksi jäsenrekisterin suunnitteluun laskutuksen kannalta toteutukseen vaikuttivat Finanssialan Keskusliiton ohjeistukset ja säännöt laskujen ulkoasusta ja laskujen sisältämistä tiedoista.

4.2.1 Yhdistyslaki (503/1989)

Yhdistykseen ja niiden toimintaan sovelletaan yhdistyslakia 503/1989. Laki velvoittaa yhdistyksen hallituksen pitämään luetteloa yhdistyksen jäsenistä. Lain mukaan luetteloon on merkittävä jokaisen jäsenen täydellinen nimi sekä kotipaikka. (Finlex 1989: L 503 3:11).

4.2.2 Henkilötietolaki (523/1999)

Yhdistyksen jäsenten henkilötietojen rekisteröintiä ohjaa henkilötietolaki (523/1999), jonka tehtävänä on ylläpitää yksityiselämän suojaa ja edistää hyvää tietojenkäsittelytapaa. Lakia sovelletaan, kun henkilötiedot muodostavat henkilörekisterin tai sen osan ja sitä sovelletaan sekä automaattiseen että manuaaliseen tietojenkäsittelyyn.

Henkilötietolain mukaan henkilötietojen käsittelyä koskevat seuraavat yleisperiaatteet:

- huolellisuusvelvoite
- henkilötietojen käsittelyn suunnittelu
- käyttötarkoitussidonnaisuus (yhteysvaatimus)
- henkilötietojen käsittelyn yleiset edellytykset (tarpeellisuusvaatimus)
- tietojen laatua koskevat periaatteet (virheettömyysvaatimus). (Finlex 1999: L 523 2:5-9).

4.2.3 Rekisterinpitäjän velvollisuudet

Henkilötietolain huolellisuusvelvoitteen mukaan henkilötietoja on käsiteltävä laillisesti hyvää tiedonhallintatapaa noudattaen. Henkilötietoja saa käsitellä ainoastaan rekisteröidyn suostumuksella ja rekisteröidyllä on oltava asiallinen yhteys, kuten yhdistyksen jäsenyys, rekisterin ylläpitäjään (yhteysvaatimus).

Henkilötietojen käsittelyn on oltava myös suunnitelmallista ja perusteltua rekisterinpitäjän toiminnan kannalta. Henkilötietojen käsittelyn tarkoitus on määriteltävä niin, että siitä käy ilmi, millaisten tehtävien hoitamiseksi henkilötietoja kerätään ja käsitellään. Käsiteltävien henkilötietojen tulee olla määritellyn tarkoituksen kannalta tarpeellisia (tarpeellisuusvaatimus).

Rekisterinpitäjän velvollisuuksiin kuuluvat mm.

- rekisteriselosteen teko ja ylläpito
- henkilötietojen turvallinen säilytys
- vaitiolovelvollisuus
- tietojen turvallinen hävitys, mikäli ne eivät enää ole rekisterinpitäjän toiminnan kannalta tarpeen mukaisia
- virheellisten, epätäydellisten tai vanhentuneiden tietojen korjaus (virheettömyysvaatimus). (Finlex 1999: L 523 2:9-10, 7:32-34).

Henkilörekisterin tiedot on suojattava siten, ettei asiaan kuulumattomilla ole pääsyä tietoihin. Laissa edellytetään, että rekisterinpitäjä toteuttaa tarpeelliset teknisesti ja organisatoriset toimenpiteet tietojen suojaamiseksi. Tarpeellisia toimenpiteitä tietojen suojaamiseksi ei ole tarkemmin määritelty, vaan lähtökohtaisesti rekisterinpitäjän on itse arvioitava riittävä tietoturvan taso. Riittävään tietoturvasoon vaikuttavat rekisteriin kerättävät tiedot sekä rekisterin laajuus. Vanhentuneet tai rekisterinpitäjän kannalta merkityksettömiksi käyneet henkilötiedot on myös hävitettävä turvallisesti, ilman, että ne joutuvat ulkopuolisten käsiin. Rekisterinpitäjää koskee myös vaitiolovelvollisuus.

4.2.4 Rekisteröidyn oikeudet

Henkilötietolain mukaan rekisteröidyn henkilön oikeuksiin kuuluvat:

- tiedon saantioikeus
- tarkastusoikeus
- tiedon korjaaminen
- kielto-oikeus. (Finlex 1999: L 523 9:39).

Tiedon saantioikeuden mukaan rekisteröidyllä on oikeus tietää tiedonkäsittelystä, sen tarkoituksesta ja tietojen luovutuksesta. Tiedot on annettava rekisteröidylle, kun niitä kerätään tai tallennetaan tai jos tiedot hankitaan muualta kuin rekisteröidyltä itseltään.

Tarkastusoikeuden mukaan rekisteröidyllä on oikeus saada tietää mitä tietoja hänestä on henkilörekisteriin tallennettu tai ettei rekisteriin ole tallennettu häntä koskevia tietoja. Lisäksi rekisteröidyllä on oikeus tarkastaa omat tietonsa rekisteristä erikseen sitä pyydettyäessä. Tietojen tarkastuksen on oltava maksutonta kerran vuodessa.

Rekisteröidyllä on oikeus korjauttaa virheellinen tai puutteellinen henkilötieto. Lisäksi rekisteröidyllä on oikeus pyytää poistamaan vanhentunut tai käsittelyn kannalta tarpeeton tieto rekisteristä.

Rekisteröidyllä on myös oikeus kieltää käsittelemästä itseään koskevia tietoja suoramainontaa, etämyyntiä, mielipidetutkimusta tms. varten.

4.2.5 Rekisteri- ja tietosuojaseloste

Henkilötietolaki edellyttää, että rekisterinpitäjän on laadittava rekisteriseloste, joka on oltava kaikkien saatavilla. Rekisteriselosteesta on ilmevä:

- rekisterinpitäjä
- rekisterinpidosta vastaava henkilö ja hänen yhteystietonsa
- henkilötietojen käsittelyn tarkoitus
- rekisterin tietosisältö
- rekisterin suojauksen periaatteet (Tietosuojavaltuutetun toimisto 2012).

Kotkan Melojen jäsenrekisteriä varten laadittiin tietosuojavaltuutetun ohjeistuksen mukainen tietosuojaseloste, joka poikkeaa rekisteriselosteesta ainoastaan siinä, että se antaa rekisteröidylle lisätietoja rekisteröidyn oikeuksista. Alla olevassa kuvassa (

Kuva 6) on yhdistyksen jäsenrekisterin tietosuojaseloste.

TIETOSUOJASELOSTE

Henkilötietolaki (523/99) 10 § Laatimispäivä: 20.6.2012	
Rekisterin pitäjä:	Kotkan Melojat ry
Yhteyshenkilö:	Jonna Lahtinen (sihteeri)
Yhteystiedot:	sihteeri@kotkanmelojat.net
Rekisterin nimi:	Kotkan Melojat ry:n jäsenrekisteri
Henkilötietojen käsittelyn tarkoitus:	Rekisterinpitäjä käsittelee asiakkaiden henkilötietoja seuraavia tarkoituksia varten: jäsenkirjeiden ja -tiedotteiden lähetyks jäsen-, kalusto-, säilytys- ym. maksujen teko, määrittely ja lähetyks maksujen valvonta palvelujen tarjoaminen jäsenille
Rekisterin tietosisältö:	Rekisteriin kerättävät tiedot: Etu- ja sukunimi * Syntymävuosi (alle 16v. alennettu jäsenmaksu) Osoitetiedot (laskujen, tiedotteiden ja jäsenkirjeiden lähetyks varten) Puhelinnumero Sähköpostiosoite ** Liittymisvuosi (statuksen määrittely) Status (maksujen määrittelyä varten esim. vapaajäsenillä ei jäsenmaksuja) Lisätiedot (valinnaisia lisätietoja tai rekisterin ylläpitoon liittyviä tietoja) Laskutustiedot (määrittämään mitä laskuja kenellekin kuuluu)
Säännön mukaiset tietolähteet:	Tiedot kerätään jäsenhakemuksista. Tietoja päivitetään jäsenten omasta pyynnöstä tai jäsenet päivittävät itse omat tietonsa.
Tietojen säännön mukaiset luovutukset:	Tietoja ei luovuteta eteenpäin kolmansille osapuolille.
Rekisterin suojausten periaatteet:	Jäsenrekisteri on suojattu ulkopuoliselta käytöltä ja kaikki rekisteriin kerätty tieto säilytetään niin, että vain jäsenasioita käsittelevillä henkilöillä on pääsy rekisteriin. Jäsenillä on henkilökohtaiset käyttäjätunnukset ja salasana järjestelmään ja jäsenillä on pääsy ainoastaan omiin tietoihinsa.
Rekisteröidyn kiello-oikeus:	Rekisteröity voi ilmoittaa luovutus- tai käsittelykiellosta kirjallisesti tai puhelimitse rekisteriä hoitavalle henkilölle.
Rekisteröidyn tarkastusoikeus:	Henkilötietolain 26 §:n mukaan rekisteröidyllä on oikeus tarkastaa itseään koskevat rekisteritiedot ja saada niistä pyydettyä kopiota. Tarkastus on maksutonta kerran vuodessa.
Tiedon korjaaminen:	Henkilötietolain 29 §:n mukaan rekisterinpitäjän on ilman aiheetonta viivytystä oma-aloitteisesti tai rekisteröidyn vaatimuksesta oikaistava, poistettava tai täydennettävä rekisterissä oleva, käsittelyn tarkoituksen kannalta virheellinen, tarpeeton tai vanhentunut henkilötieto. Jos tiedon korjaamisesta tai tarkastusoikeudesta kieltäydytään, jäsenelle annetaan kirjallinen kieltäytymistodistus ja hänellä on oikeus saattaa asia tietosuojavalvutetun ratkaistavaksi osoitteella: Tietosuojavalvutetun toimisto PL 315 00181 Helsinki Tietosuojavalvutettu voi antaa rekisterinpitäjälle määräyksen tiedon korjaamisesta. Tarkastusoikeuden toteuttamiseksi ja tiedon korjaamiseksi asianomaisen tulee ottaa yhteyttä rekisteriä hoitavaan henkilöön.
* pakollinen tieto ** pakollinen, jos jäsenkirje halutaan sähköpostiin	

Kuva 6 Tietosuojaseloste.

4.2.6 Hyvä tiedonhallintatapa

Hyvä tiedonhallintatapa on määritelty alun perin viranomaisten toimintaan, mutta sen sisältämiä elementtejä käytetään yleisemminkin tiedon hallinnassa. Hyvä tiedonhallintatavan määrittelyn lähtökohtina ovat olleet laki viranomaisen toiminnan julkisuudesta (621/1999), arkistolaki (831/1994) ja henkilötietolaki (523/1999). Hyvä tiedonhallintatapa koskee asiakirjojen ja tietojen käsittelyä ja niihin liittyvää tietoturvaa. Valtiovarainministeriön mukaan:

Hyvä tiedonhallintatapa on toimintatapa, johon kuuluu toiminnan korkea taso ja hyvä laatu. Hyvän laadun vaatimus koskee ennen kaikkea hallinnon käsittelemiä asiakirjoja ja tietoja. Näiltä vaadittavia ominaisuuksia ovat käytettävyys ja saatavuus, eheys ja virheettömyys sekä luottamuksellisuus. Asiakirjojen ja tietojen laatu varmistetaan hallintotoiminnan ja tietojärjestelmien avulla. (Valtiovarainministeriö 2002).

Hyvän tiedonhallinnan rinnalla on aiemmin puhuttu myös hyvästä rekisteritavasta, joka myöhemmin on liitetty osaksi hyvää tiedonhallintatapaan. Hyvä rekisteritapa ottaa kantaa mm. henkilömatrikkelin perustamisen edellytyksiin, rekisteriselosteisiin, tiedotukseen, mitä tietoja saa kerätä ja mistä tiedot saadaan. Tietojen keräyksestä rekisteriin tietosuojavaltuutettu antaa seuraavan ohjeen:

Hyvä rekisteritapa edellyttää, että pääosa matrikkelirekisterin tiedoista kerätään henkilöltä itseltään. Näin voidaan parhaiten varmistua tietojen oikeellisuudesta sekä siitä, ettei rekisteröity halua käyttää lain hänelle suomaa oikeutta kieltää tietojensa käyttäminen. (Tietosuojavaltuutetun toimisto 2010: 3).

Käytännössä hyvä tiedonhallintatapa tarkoittaa seuraavien asioiden huomioimista:

- arkistonmuodostussuunnitelman päivittäminen ja tarkistaminen
- tietojärjestelmä- ja henkilörekisteriselosteiden laadinta
- tietojärjestelmien ominaisuuksien suunnittelu/tarkistaminen
- tietoturvasuunnitelma
- asiakirjojen ja tietojen julkisuuden määrittely
- asiakirjojen ja tietojen suojelu

- viestinnän suunnittelu (Arkistolaitos 2004: 7)

4.2.7 Tilisiirto

Suomi siirtyi yhtenäiseen euromaksualueeseen SEPA:an (Single Euro Payments Area) vuonna 2008. SEPA-yhteensopivien tilisiirtojen teko on ollut mahdollista saman vuoden alusta asti. Finanssialan Keskusliitto määrittelee kotimaan maksuliikenteessä käytettävän tilisiirtolomakkeen tekniset yksityiskohdat ja se on laatinut ohjeet SEPA-alueella käytettävistä tilisiirtolomakkeista ja niiden rakenteesta. Ohjeen mukainen tilisiirtolomake on otettava käyttöön viimeistään 31.12.2012. Uusien tilisiirtostandardien ja sääntöjen myötä tilisiirtolomakkeisiin tuli uusia teknisiä vaatimuksia kuten IBAN-tilinumeron (International Bank Account Number) pakollisuus sekä laskuttajan pankin BIC koodi (Bank Identifier Code), joka on pankin yksilöivä tunniste. Finanssialan Keskusliiton ohje määrittelee mm. lomakkeella käytettävät värit, fontit, viivapaksuudet ja kirjainkoot. Lisäksi määritellään lomakkeen mitat, käytettävä paperi ja painoväri, tilisiirron osat ja vähimmäistietosisältö.

SEPA End Date -asetus on Euroopan parlamentin ja neuvoston asetus euromääräisiä tilisiirtoja ja suoraveloituksia koskevista teknisistä ja liiketoimintaa koskevista vaatimuksista. Se määrittää mm. takarajat kansallisista maksupalveluista luopumiselle.

Komission on antanut euromääräisiä tilisiirtoja ja suoraveloituksia koskevan asetuksen jonka tarkoituksena on varmistaa SEPA-tilisiirtoon ja suoraveloitukseen siirtyminen jäsenvaltioissa. Asetus on hyväksytty EU:n parlamentissa 14.2.2012 ja julkaistu 30.3.2012. Asetuksessa säädetään siirtymäajasta, jonka päättymiseen mennessä kansallisista tilisiirtoja ja suoraveloituspalveluista on luovuttava. Sääntely koskee lisäksi teknisiä vaatimuksia, kuten IBAN-tilinumeron pakollisuutta, tilisiirron ja suoraveloituksen vähimmäistietosisältöä, suoraveloitusta koskevia tarkistusvaatimuksia ja ISO 20022 -pohjaisten standardien käyttöönottoa myös pankin ja pankin asiakkaan välisessä maksuaineistojen välityksessä. (Finanssialan Keskusliitto 2012).

4.2.8 Viitenumero

Viitenumeroa käytetään yksilöimään yksi tietty tilisiirto. Kotkan Melojat ry:n jäsenrekisteriin luotiin algoritmi, joka laskee viitenumerot automaattisesti perustuen tiettyyn alkulukuun sekä rekisterissä olevan jäsenen jäsennumeroon. Jäsenrekisteri käyttää suomalaista viitenumerojärjestelmää, jonka käyttö on sallittua toistaiseksi.

Finanssialan Keskusliitto (Finanssialan Keskusliitto 2009) antaa ohjeet viitenumeron laskemiseksi. Viitenumero voidaan muodostaa vapaasti perustuen esimerkiksi asiakasnumeroon. Viitenumeroksi suositellaan käytettävän mahdollisimman lyhyttä numerosarjaa näppäilyvirheiden välttämiseksi, mutta sen on oltava kuitenkin vähintään 4-numeroinen. Viitenumero lasketaan kertomalla perustiivisteeksi valitun luvun numerot oikealta vasemmalle painoarvoilla 7, 3 ja 1. Tulot lasketaan yhteen ja summa vähennetään seuraavasta täydestä kymmenluvusta. Saatu erotus on viitenumeron tarkistelukku, jonka tarkoituksena on estää tallennusvirheet. Tarkistelukku lisätään viitenumeron viimeiseksi luvuksi. Jos erotukseksi saadaan tasan 10, on tarkistenumero 0. Viitenumeron etunollia ei tulosteta tilisiirtolomakkeelle ja viite on ryhmiteltävä viiden numeron ryhmiin.

4.2.9 Sähköpostilasku

Esitutkimusvaiheessa kävi ilmi tarve lähettää sähköpostilaskuja järjestelmän kautta. Projektin edetessä sähköpostilaskutus päätettiin kuitenkin siirtää jakokehityskohteeksi. Sähköpostilasku on laskun vastaanottotapa eli se ei ole sama kuin verkkolasku.

Verkkolaskun perusajatuksena on, että lasku toimitetaan ja maksetaan paperittomasti. Verkkolasku lähetetään kuluttajan valitsemaan palveluun, josta on suora yhteys sähköiseen maksamiseen verkkopankissa. Sitä esimerkiksi sähköpostin liitteenä lähetetty lasku, joka maksetaan samalla tavalla kuin paperilasku, ei ole verkkolasku (Kuluttajavirasto 2008).

Sähköpostilasku voidaan maksaa pankin konttorissa tai verkkopankissa. Se on myös ympäristöystävällinen tapa ja se vähentää lähetyskuluja ja paperin määrää. Sähköpostilaskuja voidaan kuluttajaviraston määräyksestä lähettää kuitenkin ainoastaan jäsenen omalla suostumuksella:

Kuluttajan kanssa on selkeästi sovittava käytettävästä laskutustavasta ja kuluttajan vaihtoehtoista laskun maksamisesta. Esimerkiksi kuluttajan sähköpostiosoitteen välittyminen kaupanteon yhteydessä ei riitä perusteeksi lähettää hänelle lasku sähköpostilla, vaan tällaisesta laskutustavasta pitää nimenomaisesti sopia (Kuluttajavirasto 2008).

5 MÄÄRITTELY- JA SUUNNITTELUVAIHE

Määrittely- ja suunnitteluvaihe jakautuu periaatteessa kolmeen osaan: sidosryhmien vaatimusten määrittelyyn (vaatimusmäärittely), ei-toiminnallisten vaatimusten määrittelyyn sekä käytettävien alustojen ja tekniikoiden määrittelyyn (tekninen määrittely). Vaiheen tarkoituksena on selvittää mitä lopullinen järjestelmä tekee ja miten ja millä se toteutetaan.

Suurena osana määrittely- ja suunnitteluvaiheeseen kuului myös koko projektin toteutuksen ja hallinnan suunnittelu. Kuten aiemmin on mainittu, yhdistyksen toiminta perustuu talkootoimintaan ja vapaaehtoisuuteen, joten erillisiä suunnitelmia projektibudjetista tai henkilöresursseista ei katsottu tarpeellisiksi laatia. Myöskään erityistä projektiorganisaatiota ei sinänsä ollut, vaan yhdistyksen puheenjohtaja, rahastonhoitaja ja jäsenet osallistuivat projektiin tarpeen ja omien mahdollisuuksien mukaan.

Määrittely- ja suunnitteluvaihe aloitettiin selvittämällä mitä jäsenrekisteriltä vaaditaan, jotta sen avulla saataisiin ratkaistua olemassa olevat ongelmat ja voitaisiin tehostaa ja parantaa toimintaa. Eri toiminnot listattiin käyttäjäryhmittäin ja suunniteltiin niiden toteutusta. Sidosryhmien vaatimuksia selvitettiin ja tarkennettiin mm. keskusteluilla ja erillisillä kyselyillä. Toiminnoista piirrettiin periaatteelliset näyttökuvat (vrt. 3.5.3 Prototyypimalli s. 19), joiden avulla rekisterisovelluksen toimintaa mallinnettiin sidosryhmille. Näyttökuvien ja rautalankamallien avulla pyrittiin vakauttamaan sidosryhmien vaatimuksia ja torjumaan väärinymmärryksiä. Toiminnallisen määrittelyn tuloksena saatiin suunnitelmat käyttöliittymästä ja sen toiminnoista, käyttäjäryhmistä ja tietovarastoon tarvittavista ja kerättävistä tiedoista.

Toiminnallinen ja tekninen määrittely tehtiin osittain samanaikaisesti. Seuraavissa kappaleissa on kerrottu tarkemmin määrittely- ja suunnitteluvaiheen tuloksista.

5.1 Toiminnallinen määrittely

Jäsenrekisterin sidosryhmiksi tunnistettiin kolme eri ryhmää (kts. 4.1.7 Sidosryhmät s. 4.1.7): hallituksen jäsenet, jäsenet ja muut yhdistyksen toiminnasta kiinnostuneet. Koska sidosryhmien tarpeet poikkesivat toisistaan oleellisesti, päätettiin Kotkan Melojat ry:n jäsenrekisteriin luoda kaksi erilaista käyttäjäryhmää, joiden pääsy eri toimintoihin erotetaan kirjautumisvaiheessa. Kun palveluun luodaan uudet käyttäjätunnukset, määritellään samalla kumpaan käyttäjäryhmään tunnukset kuuluvat. Kirjaututtaessa käyttäjäryhmille näytetään erilaiset valikot ja toiminnot sen mukaan mihin keneläkin on oikeudet.

Kotkan Melojat ry:n hallitukselle luotiin käyttäjäryhmä, jolla on ylläpitäjän oikeudet rekisterin toimintoihin. Ylläpitäjillä on kaikki oikeudet rekisteriin. Toinen käyttäjäryhmä on tarkoitettu yhdistyksen jäsenille. Jäsenillä on oikeus muuttaa omia tietojaan, lähettää palautetta ylläpidolle ja lähettää viestejä jäsenille tarkoitetulle keskustelupalstalle. Alla olevasta taulukosta (Taulukko 1) käyvät tarkemmin selville eri käyttäjäryhmien oikeudet.

Taulukko 1 Käyttäjäryhmien oikeudet toiminnoittain.

Toiminto	Käyttäjärühmät		
	Ylläpito	Jäsen	Tunnuksettomat käyttäjät
Jäsenten listaus	X		
Uusien jäsentietojen lisäys, poisto ja muokkaus	X		*)
Omien tietojen muokkaus	X	X **)	
Jäsenten etsiminen hakuehdoilla	X		
Jäsenkirjeiden luonti poisto ja muokkaus	X		
Laskujen teko	X		
Laskuraportit	X		
Käyttäjätunnusten luonti	X		***)
Käyttäjätunnusten poisto ja muokkaus	X		
Salasanan muuttaminen	X	X	
Käyttäjätunnuksen muuttaminen	X		
Palautelaatikon luku, viestien poisto ja viesteihin vastaaminen	X		
Oman salasanan vaihto	X	X	
Palautteen lähetyk		X	
Keskustelupalstan luku ja viestien lähetyk	X	X	
Keskustelupalstan viestien poisto	X		
Jäsenhakemus	X	X	X
Rekisteriseloste	X	X	X

*) Jäsenhakemukseen täytettävät tiedot tallennetaan suoraan tietokantaan.

**) Jäsenillä on oikeus muuttaa omia yhteystietojaan, muttei maksutietojaan tai statustaan.

***)) Jäsenhakemus luo käyttäjätunnuksen ja generoi salasanan automaattisesti, mikäli sähköpostiosoitteenttä on täytettynä. Ennen kuin hakemuksen täyttäjä on virallisesti hyväksytty seuran jäseneksi, ei tunnuksilla ole pääsyä muihin toimintoihin, kun omien tietojen muokkaukseen.

Seuraavissa kappaleissa on kerrottu tarkemmin jäsenrekisterin eri toiminnoista ja niiden määrittelyistä käyttäjärühmittäin.

5.2 Tunnuksettomat käyttäjät


Tunnuksettomilla käyttäjillä eli yhdistyksen toiminnasta kiinnostuneilla tai seuran jäseneksi haluavilla on oikeus seuran julkisten kotisivujen toimintoihin (kts.4.1.2 Kotkan Melojat ry:n kotisivut s. 23). Tunnuksettomat käyttäjät voivat halutessaan täyttää ja lähettää jäsenhakemuksen yhdistyksen kotisivujen kautta, joilta löytyy linkki hakemukseen. Koska jäsenhakemuksen tiedot menevät suoraan tietokantaan, sivulta löytyy myös linkki rekisterin tietosuojaselosteeseen (kts. 4.2.5 Rekisteri- ja tietosuojaseloste s. 30).

Kun sähköinen jäsenhakemus täytetään ja lähetetään eteenpäin, tiedot tallennetaan suoraan tietokantaan. Uuden jäsenhakemuksen statukseksi ylläpitopuolella merkitään ”Uusi hakemus”, josta hallituksen jäsenet näkevät, että hakemus on uusi, eikä jäsentä ole vielä virallisesti hyväksytty seuran jäseneksi. Jäsenhakemukseen täytettävät tiedot on esitelty alla olevassa taulukossa (Taulukko 2).

Taulukko 2 Jäsenhakemuksen tiedot selityksineen.

Etunimi	Pakollinen tieto
Sukunimi	Pakollinen tieto
Syntymäaika	Alle 16 vuotiailla on alennettu jäsenmaksu. Ilmoittamalla syntymävuotensa jäsenrekisteri laskee automaattisesti alennetun jäsenmaksun laskutuksen yhteydessä.
Katuosoite	Osoitetietoja käytetään laskujen ja jäsenkirjeiden lähetyksessä sekä osoitetarrojen tulostuksessa kirjeisiin.
Postinumero	
Postitoimipaikka	
Puhelinnumero	Käytetään tarvittaessa yhteydenpidossa jäseniin.
Sähköpostiosoite	Sähköpostiin lähetetään salasana ja käyttäjätunnukset, jos se on mainittu jäsentiedoissa. Jos hakemuksen täyttäjällä haluaa jäsenkirjeen sähköpostiin, tämä kenttä on pakollinen.
Jäsenkirje: sähköpostiin / postiin	Jäsen voi valita haluaako hän jäsenkirjeensä sähköpostiin vai kirjeitse. Jos valitaan sähköpostiin, sähköpostiosoitte kenttä on pakollinen tieto.
Jäsenmaksutiedot	Seuran jäsenet maksavat jäsenmaksun paitsi, jos jäsen on vapaajäsen tai kuuluu seuran hallitukseen. Alle 16 vuotiailla on alennettu jäsenmaksu, joka lasketaan automaattisesti syntymävuodesta.
Kaluston käyttömaksu	Jos jäsen haluaa käyttää seuran kalustoa, on hänen maksettava kaluston käyttömaksu.
Kaluston säilytysmaksut	Säilytysmaksu on maksettava, jos jäsen haluaa säilyttää omaa kalustoaan seuran vajoilla. Erilaisille kajakeille on omat säilytysmaksunsa, jotka määräytyvät lähinnä kaluston viemän tilan mukaan.
Lisätiedot	Lisätietoihin voidaan kirjoittaa vapaamuotoista tekstiä esim. aiemmasta melontakokemuksesta tai tietoja osallistumisesta seuran järjestämiin tapahtumiin, kuten alkeiskurssiin. Seuran jäseneksi pääseminen edellyttää näyttöä seuran toimintaan osallistumisesta.

Alla olevassa kuvassa (Kuva 7) on näyttökuva yhdistyksen jäsenhakemuksesta.


**KOTKAN
MELOJAT RY**

JÄSENHAKEMUS

[Takaisin Kotkan Melojien sivuille...](#)
[Rekisteriseloste](#)

TÄYTÄ JÄSENHAKEMUS

Täytä alla olevat tiedot ja lähetä ne eteenpäin "Lähetä hakemus" -painikkeella. Onnistuneen lähetyksen jälkeen saat tulostusversion hakemuksestasi.

* merkityt kentät ovat pakollisia (*) merkitty kenttä pakollinen, jos valittu "jäsenkirje sähköpostiin"

PERUSTIEDOT

Etunimi: * Sukunimi: * Syntymävuosi: ---Valitse---

Katuosoite:

Postinumero: Postitoimipaikka:

Puhelinnumero:

Sähköpostiosoite: *

Jäsenkirje: ☐ sähköpostiin ☐ postiin

LASKUTUSTIEDOT

Liityn jäseneksi ja maksan jäsenmaksun (25€; alle 16 v. 15€): ☐

Ei jäsenmaksua: ☐

Maksan kaluston käyttömaksun (50€ rastita, jos haluat käyttää seuran välineitä): ☐

OMAN KALUSTON SÄILYTYS

Maksan kaluston säilytysmaksun (jos haluat säilyttää vajassa omia välineitä)

Metsola: Retkikajakit, kaksikot (20€/kajakki): 0

Metsola: Poolokajakit, koskarit (10€/kajakki): 0

Metsola: Intiaanikanootti (30€/kanootti): 0

Mansikkalahti: Retkikajakit, kaksikot (20€/kajakki): 0

Mansikkalahti: Poolokajakit, koskarit (10€/kajakki): 0

Mansikkalahti: Intiaanikanootti (30€/kanootti): 0

MUUT TIEDOT

Lisätietoja:

KoMe ry – jäsenrekisteri 2011–2012

Kuva 7 Jäsenhakemus.

Jos hakemuksen sähköpostikenttä on täytettynä, hakemuksessa ilmoitettuun sähköpostiin lähetetään vahvistus tietojen perille menosta. Hakemus voidaan haluttaessa myös tulostaa paperille. Jos sähköposti on täytetty, jäsenrekisteri generoi automaattisesti myös käyttäjätunnuksen (sähköpostiosoite) sekä salasanan palveluun, jotka lähetetään hakemuksen mukana sähköpostiin. Koska käyttäjätunnukset voi näin ollen saada kuka tahansa, ei tunnuksilla ole pääsyä muualle kuin omien tietojen muutokseen. Vasta kun hakemus on virallisesti hyväksytty, avataan käyttäjätunnuksille pääsy myös keskuste-

lupalstalle ja palautelaatikkoon. Alla olevassa kuvassa (Kuva 8) on ilmoitus hakemuksen vastaanottamisesta lähetyksen jälkeen.

KOTKAN MELOJAT RY **JÄSENHAKEMUS**

[Takaisin Kotkan Melojien sivuille...](#)
[Rekisteriseloste](#)

JÄSENHAKEMUS

Jäsentietojen tallennus onnistui.
Jäsenhakemus on lähetetty sähköpostiisi.

PERUSTIEDOT

Nimi: Mallikas, Matti Syntymävuosi: 1963
Osoite: Merikatu 8, 48100 Kotka
Puhelin: 010 123 1234
Sähköposti: matti.mallikas@melojat.fi
Jäsenkirje: kirje sähköpostiin

MAKSUT:

Jäsenmaksu: X
Kalustomaksu: X

Säilytysmaksut Metsola:
Säilytysmaksut Mansikkalahti:

Lisätietoja: Osallistuin tänä keväänä järjestetyille alkeiskursseille.

[Tulosta lomake](#)

KoMe ry - jäsenrekisteri 2011-2012

Kuva 8 Kuittausilmoitus jäsenhakemuksen vastaanotosta.


Koska kaikki jäsenet tai jäseneksi haluavat eivät käytä internetiä, säilytetään jäsenhakemuksesta myös paperiversio, joita saa seuran vajalta.

5.3 Jäsentason käyttäjät

Seuraavissa kappaleissa on tarkemmin kuvattuina jäsentason käyttäjille suunnitellut toiminnot.

5.3.1 Omien tietojen muokkaus

Jäsentason käyttäjillä on oikeus nähdä ja muokata omia tietojaan. Toiminnon avulla jäsenrekisterin tiedot saadaan pysymään paremmin ajan tasalla, kun jokainen voi tarkistaa omat yhteystietonsa. Alla olevassa kuvassa (Kuva 9) on esitelty omien tietojen muokkaus-ikkuna. Poikkeuksina ylläpitotason jäsentietojen muokkausikkunaan, lomakkeella ei voida muuttaa seuraan liittymisvuosi-, status- tai lisätietokenttiä.


**KOTKAN
MELOJAT RY**

JÄSENREKISTERI

Kirjautuneena: matti.mallikas@melojat.net [Kirjaudu ulos](#)

JÄSENREKISTERI
Omat tiedot
[Vaihda salasana](#)
[Keskustelupalsta](#)
[Lähetä palautetta](#)
[Rekisteriseloste](#)

JÄSENTIETOJEN MUOKKAUS

Etunimi: Sukunimi: Syntymävuosi:
Katuosoite:
Postinumero: Postitoimipaikka:
Puhelinnumero:
Sähköpostiosoite: (*)
Jäsenkirje: ☒ sähköpostiin ☐ postiin

LASKUTUSTIEDOT

Jäsenmaksu (25€; alle 16 v. 15€): ☒ Ei jäsenmaksua: ☐
Kaluston käyttömaksu (50€): ☒

OMAN KALUSTON SÄILYTYS

Kaluston säilytysmaksu (jos haluat säilyttää vajassa omia välineitä)

Metsola: Retkikajakit, kaksikot (20€/kajakki): kpl
Metsola: Poolokajakit, koskarit (10€/kajakki): kpl
Metsola: Intiaanikanootti (30€/kanootti):
Mansikkalahti: Retkikajakit, kaksikot (20€/kajakki):
Mansikkalahti: Poolokajakit, koskarit (10€/kajakki):
Mansikkalahti: Intiaanikanootti (30€/kanootti):

Kuva 9 Omien tietojen muokkaus.

5.3.2 Salasanan muuttaminen: jäsentason näkymä

Jäsenillä on oikeus muuttaa oma salasanansa palveluun. Salasana voidaan muuttaa pyynnöstä myös ylläpitotason tunnuksilla. Ylläpitotason tunnuksilla voidaan muuttaa myös käyttäjän käyttäjätasoa (jäsen tai ylläpito) sekä käyttäjätunnusta. Alla olevassa kuvassa (Kuva 10) on kuva salasana vaihdosta. Salasanojen kelpoisuudesta on kerrottu tarkemmin kohdassa 5.6.3. Salasanojen turvallisuus s. 64.



KOTKAN MELOJAT RY **JÄSENREKISTERI**

Kirjautuneena: matti.mallikas@melojat.net [Kirjaudu ulos](#)

JÄSENREKISTERI

- [Omat tiedot](#)
- [Vaihda salasana](#)
- [Keskustelupalsta](#)
- [Lähetä palautetta](#)
- [Rekisteriseloste](#)

MUUTA SALASANASI

****merkityt kentät ovat pakollisia, jos joku niistä on täytetty. (Huom! Salasanaa ei muuteta, jos kenttiä ei ole täytetty)**

Uusi salasana: **

Salasana uudestaan: **

Kuva 10 Salasan vaihto.

5.3.3 Palautteen lähetyks

Kirjautuneet käyttäjät voivat halutessaan lähettää palautetta jäsenrekisterin ylläpidolle (Kuva 11). Halutessaan vastauksen palautteeseensa, käyttäjä voi kirjoittaa sähköpostiosoitteensa sähköpostikenttään.



KOTKAN MELOJAT RY **JÄSENREKISTERI**

Kirjautuneena: aku [Kirjaudu ulos](#)

JÄSENREKISTERI

- [Omat tiedot](#)
- [Vaihda salasana](#)
- [Keskustelupalsta](#)
- [Lähetä palautetta](#)
- [Rekisteriseloste](#)

ANNA PALAUTETTA

Viestisi:

Kirjoita sähköpostiosoitteesi, jos haluat saada vastauksen:

Kuva 11 Palautteen lähetyks.

5.3.4 Keskustelupalsta, jäsentason näkymä

Jäsenrekisterin yhteyteen luotiin keskustelupalsta, johon on pääsy ainoastaan käyttäjätunnukset omaavilla, yhdistykseen virallisesti hyväksytyillä jäsenillä. Alla olevassa (Kuva 12) kuvassa on näyttökuva keskustelupalstan aloitussivusta.



Kuva 12 Keskustelupalstan aloitussivu.

Keskustelupalstalle tulevat keskustelut listataan otsikoittain ja niihin lisätään päivämäärä, jolloin keskustelu on lisätty palstalle (kts. Kuva 13). Otsikon yläpuolelle tulee ilmoitus, milloin keskusteluun on lisätty viimeisin viesti ja otsikon perässä näytetään montako viestiä keskustelu sisältää.

Viimeisin viesti: Tänään	
22.08.2012 21:08:28 Muinaistuliille Kaunissaareen	(viestit: 8 kpl)
Viimeisin viesti: 07 päivää sitten	
21.08.2012 15:08:18 Ensi talven uimahallivuorot	(viestit: 1 kpl)
Viimeisin viesti: 05 päivää sitten	
20.08.2012 19:08:15 Lähes täydenkuunmelonta 29.082012	(viestit: 3 kpl)
Viimeisin viesti: 06 päivää sitten	
20.08.2012 12:08:09 Harjoittelua	(viestit: 8 kpl)
Viimeisin viesti: 06 päivää sitten	
20.08.2012 10:08:55 Seuran retki	(viestit: 6 kpl)
Viimeisin viesti: 08 päivää sitten	
20.08.2012 10:08:09 jäsenrekisteri	(viestit: 3 kpl)

Kuva 13 Ote keskustelupalstan keskustelulistasta.

Osallistuakseen keskustelupalstan keskusteluihin, kirjautuneen käyttäjän on ensin luotava itselleen nimimerkki, jolla lähetetyt viestit allekirjoitetaan. Nimimerkkiä ei voi jäsentason käyttäjän toimesta muuttaa myöhemmin. Nimimerkitön käyttäjä voi lukea muiden keskusteluja, muttei vastata viesteihin. Kun nimimerkki on luotu, keskustelupalstan aloitussivulla (Kuva 12) oleva ”Aloita uusi keskustelu” -linkki muuttuu aktiiviseksi, ja käyttäjä voi halutessaan aloittaa uuden aiheen (Kuva 14).



KOTKAN MELOJAT RY **JÄSENREKISTERI**

Kirjautuneena: aku [Kirjaudu ulos](#)

JÄSENREKISTERI

- [Omat tiedot](#)
- [Vaihda salasana](#)
- Keskustelupalsta
- [Lähetä palautetta](#)
- [Rekisteriseloste](#)

Uusi keskusteluaihe

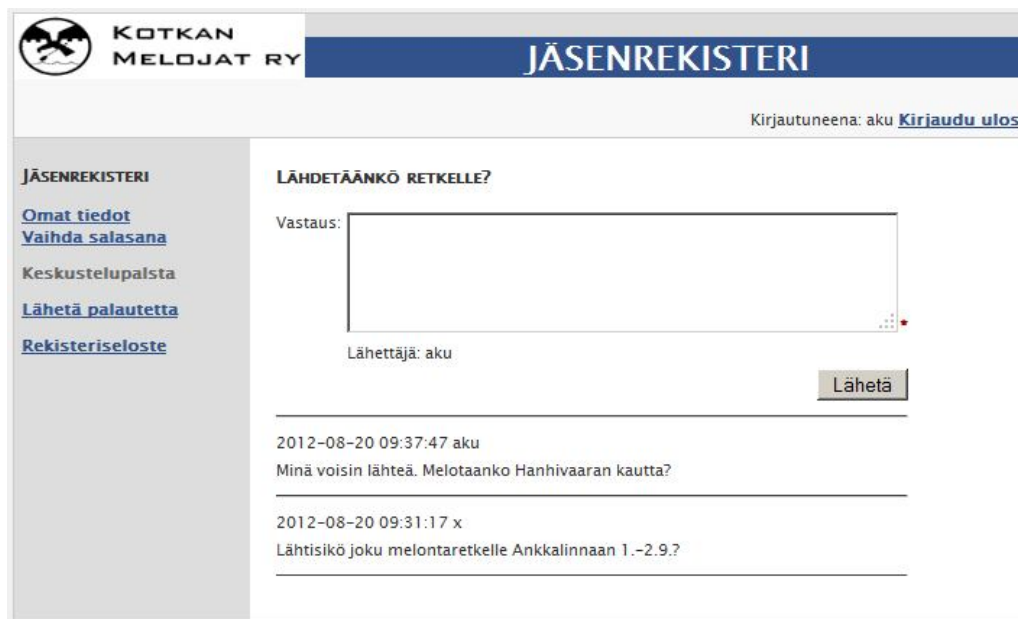
Otsikko:

Viesti:

Lähtetty: aku

Kuva 14 Uuden keskustelun aloitus.

Keskusteluja päästään lukemaan keskustelun otsikon linkistä. Nimimerkittömälle käyttäjälle ei näytetä vastauksen kirjoituskenttää, mutta keskusteluja voidaan kyllä lukea. Nimimerkillinen käyttäjä voi lisätä uuden viestin ketjuun avaamalla keskustelun linkistä ja kirjoittamalla viestinsä vastauskenttää (Kuva 15).



KOTKAN MELOJAT RY **JÄSENREKISTERI**

Kirjautuneena: aku [Kirjaudu ulos](#)

JÄSENREKISTERI

- [Omat tiedot](#)
- [Vaihda salasana](#)
- Keskustelupalsta
- [Lähetä palautetta](#)
- [Rekisteriseloste](#)

LÄHETÄÄNKÖ RETKELLE?

Vastaus:

Lähtetty: aku

2012-08-20 09:37:47 aku
Minä voisin lähteä. Melotaanko Hanhivaaran kautta?

2012-08-20 09:31:17 x
Lähtisikö joku melontaretkelle Ankkalinnaan 1.-2.9.?

Kuva 15 Uuden viestin lisääminen keskusteluun.

5.4 Ylläpitotason käyttäjät

Seuraavissa kappaleissa on esitelty ylläpitotason toimintoja. Osa näyttökuvista on otettu sovelluksen beetatestausvaiheesta, kun sovelluksessa oli jo oikeita jäsentietoja. Tästä syystä osa kuvissa olevista tiedoista on peitetty.

5.4.1 Käyttäjälistaus

Ylläpitotason käyttäjillä on oikeus luoda, muokata ja poistaa käyttäjätunnuksia. Jäsenrekisterin käyttäjät saadaan listattua aakkosjärjestyksessä sukunimen perusteella (Kuva 16) ja listauksessa käyttäjän nimi on linkkinä käyttäjätietoihin.

KOTKAN
MELOJAT RY

JÄSENREKISTERI

Kirjautuneena: jonna [Kirjaudu ulos](#)

JÄSENREKISTERI

[Näytä kaikki jäsenet](#)

[Lisää uusi jäsen](#)

[Etsi jäseniä](#)

JÄSENKIRJEET

[Lisää jäsenkirje](#)

[Näytä kirjepohjat](#)

LASKUTUS

[Tee uusi lasku](#)

[Näytä laskuraportit](#)

KÄYTTÄJÄHALLINTA

[Näytä kaikki käyttäjät](#)

[Luo uudet](#)

[käyttäjätunnukset](#)

MUUT TOIMINNOT

[Palautelaatikko](#)

[Keskustelupalsta](#)

[Rekisteriseloste](#)

KÄYTTÄJÄTIETOJEN HALLINTA

Käyttäjätaso Nimi

Käyttäjätunnus

Viimeksi kirjautunut

jasen

[\[linkki\]](#)

[peitetty]

12:03:26 25.10.2012

jasen

[\[linkki\]](#)

[peitetty]

13:28:52 24.10.2012

jasen

[\[linkki\]](#)

[peitetty]

admin

[\[linkki\]](#)

[peitetty]

17:28:56 20.08.2012

jasen

[\[linkki\]](#)

[peitetty]

20:47:01 23.10.2012

admin

[\[linkki\]](#)

[peitetty]

jasen

[\[linkki\]](#)

[peitetty]

16:44:49 18.09.2012

jasen

[\[linkki\]](#)

[peitetty]

17:50:33 21.10.2012

jasen

[\[linkki\]](#)

[peitetty]

22:44:00 29.08.2012

jasen

[\[linkki\]](#)

[peitetty]

17:10:42 22.10.2012

jasen

[\[linkki\]](#)

[peitetty]

21:54:24 20.08.2012

admin

Lahtinen Jonna

jonna

12:35:54 25.10.2012

Kuva 16 Käyttäjälistaus.

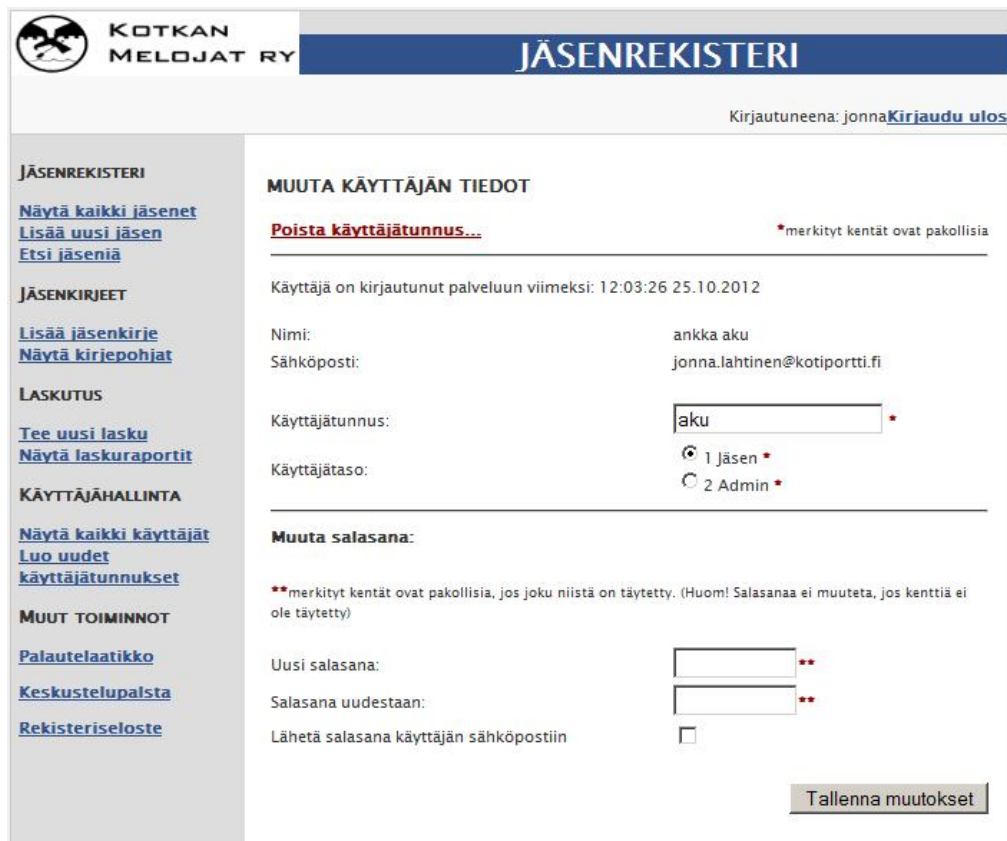
Käyttäjälistauksesta nähdään onko käyttäjä jäsen- vai ylläpitotason käyttäjä. Listauksessa on myös käyttäjän käyttäjätunnus ja milloin käyttäjä on viimeksi kirjautunut palveluun.

5.4.2 Käyttäjätunnusten luonti, muokkaus ja poisto

Uusia käyttäjätunnuksia voidaan luoda alla olevan kuvan (Kuva 17) kaltaisessa ikkunassa. Kun uusia tunnuksia lisätään, valitaan ensin alasvetovalikosta jäsen, jolle tunnuksat tehdään. Valikossa on vain niiden jäsenten nimet, joilla ei vielä ole tunnuksia palveluun. Näin estetään, ettei käyttäjälle luoda vahingossa useita tunnuksia.

Kuva 17 Käyttäjätunnusten luonti.

Käyttäjätunnukset luodaan samankaltaisessa ikkunassa kuin missä niitä muokataankin (Kuva 18). Käyttäjälle luotava käyttäjätunnus on automaattisesti jäsentiedoista haettu sähköpostiosoite, mikäli se on merkitty tai tunnus voidaan kirjoittaa suoraan kenttään. Sähköpostiosoitetta käytetään oletustunnuksena, sillä järjestelmä ei hyväksy kahta samaa käyttäjätunnusta ja sähköpostiosoite on aina uniikki. Uudelle käyttäjälle määritellään käyttäjätaso ja salasana. Salasana saadaan lähetettyä automaattisesti myös käyttäjän sähköpostiin. Käyttäjä voi myöhemmin muuttaa salasansa haluamakseen tai unohtunut salasana voidaan muuttaa ylläpitäjän toimesta alla olevan kuvan (Kuva 18) kaltaisessa ikkunassa.



KOTKAN MELOJAT RY **JÄSENREKISTERI**

Kirjautuneena: jonna [Kirjaudu ulos](#)

JÄSENREKISTERI
[Näytä kaikki jäsenet](#)
[Lisää uusi jäsen](#)
[Etsi jäseniä](#)

JÄSENKIRJEET
[Lisää jäsenkirje](#)
[Näytä kirjepohjat](#)

LASKUTUS
[Tee uusi lasku](#)
[Näytä laskuraportit](#)

KÄYTTÄJÄHALLINTA
[Näytä kaikki käyttäjät](#)
[Luo uudet käyttäjätunnukset](#)

MUUT TOIMINNOT
[Palautelaatikko](#)
[Keskustelupalsta](#)
[Rekisteriseloste](#)

MUUTA KÄYTTÄJÄN TIEDOT

[Poista käyttäjätunnus...](#) *merkityt kentät ovat pakollisia

Käyttäjä on kirjautunut palveluun viimeksi: 12:03:26 25.10.2012

Nimi: ankka aku
Sähköposti: jonna.lahtinen@kotiportti.fi

Käyttäjätunnus: *

Käyttäjätaso:
☒ 1 Jäsen *
☐ 2 Admin *

Muuta salasana:

**merkityt kentät ovat pakollisia, jos joku niistä on täytetty. (Huom! Salasanaa ei muuteta, jos kenttiä ei ole täytetty)

Uusi salasana: **

Salasana uudestaan: **

Lähetä salasana käyttäjän sähköpostiin ☐

[Tallenna muutokset](#)

Kuva 18 Käyttäjätietojen hallinta.

Käyttäjätietojen hallintaikkunassa (Kuva 18) voidaan muuttaa käyttäjän käyttäjätunnusta, salasanaa tai käyttäjätasoa. Toiminnosta löytyy myös linkki käyttäjätunnusten poistamiseksi.

5.4.3 Jäsenlistaus

Jäsenrekisterin ylläpitäjät näkevät kaikki rekisterin sisältämät jäsenet listana (Kuva 19). Jäsenten nimet on aakkostettu sukunimen perusteella. Listan yläpuolelta nähdään myös, montako jäsentä jäsenrekisterissä on.


Listauksessa jäsenen nimi on linkki jäsentietoihin. Nimen perässä on jäsenen status, joka voi olla:

- Uusi hakemus
 - o Status kertoo, että uusi jäsenhakemus on lähetetty, mutta sitä ei ole vielä käsitelty, eli hakijaa ei ole vielä virallisesti hyväksytty jäseneksi (kts. 4.1.3 Kotkan Melojat ry: jäseneksi hyväksyminen s. 24).
 - o Status on merkitty punaisella, jotta uudet hakemukset erottuisivat paremmin ja niihin voitaisiin reagoida nopeammin.
- Alkeiskurssilainen
 - o Kuluvana vuonna seuraan liittynyt jäsen, jolle ei vielä ole lähetetty jäsenmaksulaskua
 - o Statuksen avulla nähdään, montako alkeiskurssilaista on osallistunut kevään alkeiskurssille.
- Jäsen
 - o Seuran jäseneksi hyväksytty perusjäsen.

- Juniorijäsen
 - o Seuran jäseneksi valittu, alle 16 -vuotias, jäsen. Status vaikuttaa laskutukseen, sillä alle 16 -vuotiailla jäsenillä on alennettu jäsenmaksu.
- Hallituksen jäsen
 - o Hallituksen jäsenillä on oikeus saada ylläpitotason tunnukset jäsenrekisterisovellukseen.
 - o Status vaikuttaa laskutukseen, sillä hallituksen jäsenien ei tarvitse maksaa jäsenmaksuja.
- Vapaajäsen
 - o Vapaajäsenien ei tarvitse maksaa jäsenmaksuja.

5.4.4 Jäsentietojen lisäys, muokkaus ja poisto

Jäsentietojen muokkausikkuna (Kuva 20) on pienin poikkeuksin samanlainen, kuin jäsenhakemus tai jäsentason omien tietojen muutosikkunakin. Poikkeuksina ovat oikeus muuttaa jäsenen liittymisvuotta, statusta ja lisätietoja. Lisätietokenttää käytetään ylläpidon omiin merkintöihin ja siihen voidaan lisätä esim. erityisiä tietoja laskutuksesta. Lisäksi ylläpitotason käyttäjällä on oikeus poistaa jäsentiedot jäsenrekisteristä.


**KOTKAN
MELOJAT RY**

JÄSENREKISTERI

Kirjautuneena: jonna [Kirjaudu ulos](#)

JÄSENREKISTERI
[Näytä kaikki jäsenet](#)
[Lisää uusi jäsen](#)
[Etsi jäseniä](#)

JÄSENKIRJEET
[Lisää jäsenkirje](#)
[Näytä kirjepohjat](#)

LASKUTUS
[Tee uusi lasku](#)
[Näytä laskuraportit](#)

KÄYTTÄJÄHALLINTA
[Näytä kaikki käyttäjät](#)
[Luo uudet](#)
[käyttäjätunnukset](#)

MUUT TOIMINNOT
[Palautelaatikko](#)
[Keskustelupalsta](#)
[Rekisteriseloste](#)

JÄSENTIETOJEN MUOKKAUS
Poista jäsentiedot...

*merkityt kentät ovat pakollisia

Etunimi: * Sukunimi: * Syntymävuosi:

Katuosoite:

Postinumero: Postitoimipaikka:

Puhelinnumero:

Sähköpostiosoite: (*)

Jäsenkirje: ☒ sähköpostiin ☐ postiin

Liittymisvuosi: Status:

LASKUTUSTIEDOT
Alle 16v. jäsenmaksu tehdään syntymävuoden tai statuksen avulla (status=juniorijäsen).
Jäsenmaksu (25€; alle 16 v. 15€): ☒ Ei jäsenmaksua: ☐
Kaluston käyttömaksu (50€): ☒

**OMAN KALUSTON
SÄILYTYS**
Kaluston säilytysmaksu (jos haluat säilyttää vajassa omia välineitä)

Metsola: Retkikajakit, kaksikot (20€/kajakki): kpl
Metsola: Poolokajakit, koskarit (10€/kajakki): kpl
Metsola: Intiaanikanootti (30€/kanootti):

Mansikkalahti: Retkikajakit, kaksikot (20€/kajakki):

Mansikkalahti: Poolokajakit, koskarit (10€/kajakki):

Mansikkalahti: Intiaanikanootti (30€/kanootti):

Lisätietoja:

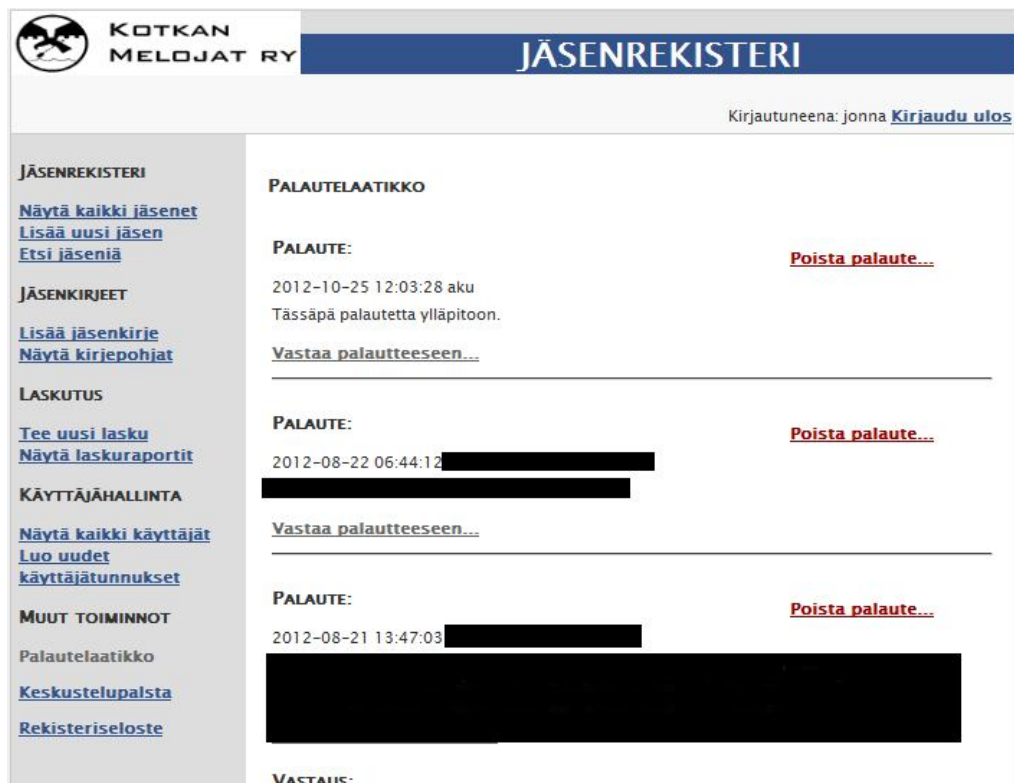
Tallenna

Kuva 20 Jäsentietojen muokkaus.

5.4.5 Palautelaatikko: ylläpitotason näkymä

Palautelaatikko on ylläpitotason käyttöliittymässä alla olevan kuvan (Kuva 21) kaltainen. Palautteet listataan aikajärjestyksessä ja listauksessa näkyvät palautteen lähettäjä sekä palautteen viesti.

Jos palautteen lähettäjä on halunnut vastuksen viestiinsä, eli on lisännyt sähköpostiosoitteensa viestin mukaan (kts.5.3.3 Palautteen lähetys s. 42), palautteen alla tulostettava linkki, ”Vastaa palautteeseen”, on aktiivinen. Muutoin se ei ole käytettävissä. Kun ylläpito on vastannut viestiin, vastausviesti ja lähettäjä tulevat näkyviin palautteen alle. Palautelaatikkoikkunassa voidaan myös poistaa vastaanotettuja palautteita.



Kuva 21 Palautelaatikko, ylläpidon näkymä.

5.4.6 Palautteeseen vastaaminen

Palautteeseen voidaan vastata sähköpostiohjelman kaltaisessa ikkunassa (Kuva 22). Vastausviesti lähetetään palautteen lähettäjän ilmoittamaan sähköpostiosoitteeseen. Viestin tiedoista on automaattisesti täytetty otsikko, lähettäjä ja allekirjoitus, mutta niitä voidaan tarvittaessa muuttaa kirjoittamalla tiedot suoraan tekstikenttiin. Viestin lop-

puun lisätään automaattisesti teksti: ”Tämä viesti on lähetetty Kotkan Melojat ry:n jäsenrekisteristä. Älä vastaa viestiin”.

The screenshot shows the 'JÄSENREKISTERI' (Member Register) page of Kotkan Melojat ry. The page has a blue header with the logo and title. A sidebar on the left contains navigation links for various functions like 'Näytä kaikki jäsenet', 'Lisää uusi jäsen', 'Etsi jäseniä', 'JÄSENKIRJEET', 'LASKUTUS', 'KÄYTTÄJÄHALLINTA', and 'MUUT TOIMINNOT'. The main content area is titled 'VASTAA PALAUTEeseen' (Respond to feedback). It includes a date and time stamp '2012-08-20 19:40:24'. Below this, there are sections for 'PALAUTE' (Feedback) and 'VASTAUS PALAUTEeseen' (Response to feedback). The 'VASTAUS PALAUTEeseen' section contains a form with fields for 'Otsikko:' (Subject), 'Lähettiläjä:' (Sender), 'Vastaus:' (Response), and 'Allekirjoitus:' (Signature). The 'Otsikko:' field is pre-filled with 'Palautteesi Kotkan Melojat ry jäsenrekisterissä'. The 'Lähettiläjä:' field is pre-filled with 'Kotkan Melojat ry'. The 'Vastaus:' field is empty. The 'Allekirjoitus:' field is pre-filled with 'Kotkan Melojat ry'. There is a 'Lähetä' (Send) button at the bottom right. A note at the bottom states: 'Voit korvata lähettäjän sähköpostiosoitteella. Jos kentään jätetään oletukseksi, viestiin lisätään teksti: "Tämä viesti on lähetetty Kotkan Melojat ry:n jäsenrekisteristä. Älä vastaa viestiin."'

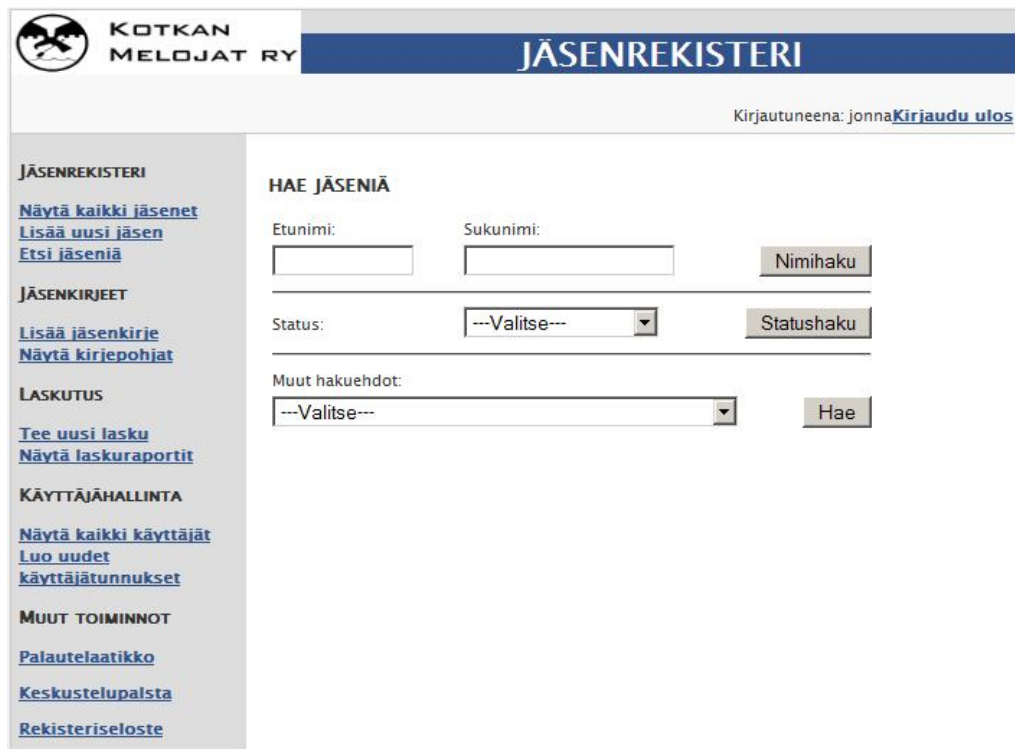
Kuva 22 Palautteeseen vastaaminen.

5.4.7 Keskustelupalstan hallinta

Keskustelupalstatoiminnot (kts. 5.3.4 Keskustelupalsta, jäsentason näkymä s. 42) ovat ylläpitopuolella samanlaiset kuin jäsentasollakin. Poikkeuksena on, että ylläpitotasolla on oikeus poistaa keskusteluja tai yksittäisiä viestejä keskusteluista.

5.4.8 Jäsentietojen haku

Jäsenrekisteristä voidaan hakea jäsentietoja erilaisten hakuehtojen perusteella (kts. Kuva 23). Hakuehdoiksi voidaan valita esimerkiksi nimi- tai statustiedot. Toimintoon on tehty valmiiksi myös muita ehtoja, joiden avulla voidaan hakea esimerkiksi jäseniä, joilla on tai ei ole sähköposti, jäseniä, jotka haluavat jäsenkirjeensä sähköpostiin. Toiminnon avulla voidaan tehdä erilaisia listauksia jäsenistä myös laskutuksen perusteella esimerkiksi lista jänistä, jotka eivät ole maksaneet jäsenmaksuja.



The screenshot shows the 'JÄSENREKISTERI' (Member Register) interface for Kotkan Melojat RY. The header includes the organization's logo and name. A navigation menu on the left lists various functions: JÄSENREKISTERI, JÄSENKIRJEET, LASKUTUS, KÄYTTÄJÄHALLINTA, and MUUT TOIMINNOT, each with sub-links. The main area is titled 'HAE JÄSENIÄ' (Search Members) and contains search criteria: 'Etunimi:' (First name) and 'Sukunimi:' (Last name) with input fields and a 'Nimihaku' (Name search) button; 'Status:' with a dropdown menu and a 'Statushaku' (Status search) button; and 'Muut hakuehdot:' (Other search criteria) with a dropdown menu and a 'Hae' (Search) button. A user login status 'Kirjautuneena: jonna' and a 'Kirjaudu ulos' (Logout) link are visible in the top right.


Kuva 23 Jäsentietojen haku.

Hakuehdot täyttävät tulokset näytetään listana (kts. Kuva 24) ja ne saadaan tarvittaessa tulostettua raportiksi. Listauksessa näytettävät tiedot riippuvat hakuehdoista.

Kuva 24 Hakutulokset.

5.4.9 Laskutus

Laskutuksen teko alkaa laskupohjan luonnilla (Kuva 25). Laskupohjalle annetaan nimi, joka kuvaa laskutusta ja jonka perusteella se tallennetaan tietokantaan. Samaa laskupohjaa voidaan käyttää myöhemmin muuttamalla laskupohjan nimeä. Esimerkiksi jos maksut pysyvät samana, voidaan vuoden 2012 jäsenmaksupohjaa käyttää vuoden 2013 laskutuksessa ilman, että uutta laskupohjaa tarvitsee luoda. Laskupohjaa ei siis tarvitse tehdä aina uudelleen, mutta jos laskutettavat maksut vaihtuvat, voidaan luoda uusi laskupohja, mutta vanha pohja on yhä tallessa.


**KOTKAN
MELOJAT RY**

JÄSENREKISTERI

Kirjautuneena: jonna [Kirjaudu ulos](#)

JÄSENREKISTERI
[Näytä kaikki jäsenet](#)
[Lisää uusi jäsen](#)
[Etsi jäseniä](#)
JÄSENKIRJEET
[Lisää jäsenkirje](#)
[Näytä kirjepohjat](#)
LASKUTUS
[Luo uusi laskupohja](#)
[Tee uusi lasku](#)
[Näytä laskuraportit](#)
KÄYTTÄJÄHALLINTA
[Näytä kaikki käyttäjät](#)
[Luo uudet käyttäjätunnukset](#)
MUUT TOIMINNOT
[Palautelaatikko](#)
[Keskustelupalsta](#)
[Rekisteriseloste](#)

LUO UUSI LASKUPOHJA
Laskupohjan nimi:
Jäsenmaksu: €
Jäsenmaksu alle 16 v: €
Kalustomaksu: €
Säilytysmaksu kajakki: €
Säilytysmaksu poolo: €
Säilytysmaksu kaksikko: €
Säilytysmaksu inkkari: €
Säilytysmaksu Mansikkalahti: €
Vajalla yöpyminen: €
Alkeiskurssi: €
Huomautuslasku: €

Kuva 25 Laskupohjan luonti.

5.4.10 Laskutuksen teko ja maksetut maksut

Laskujen luonti alkaa perustietojen, kuten laskupohjan (kts. Kuva 25), pankin, tilinumeron, eräpäivän ja maksuehdon, valinnalla (kts. Kuva 26). Pankkiyhteystiedot tulevat automaattisesti tietokannasta, mutta jos niitä on muutettava, muutokset voidaan tehdä suoraan tekstikenttiin.

KOTKAN MELOJAT RY

JÄSENREKISTERI

Kirjautuneena: jonna [Kirjaudu ulos](#)

JÄSENREKISTERI

[Näytä kaikki jäsenet](#)
[Lisää uusi jäsen](#)
[Etsi jäseniä](#)

JÄSENKIRJEET

[Lisää jäsenkirje](#)
[Näytä kirjepohjat](#)

LASKUTUS

[Luo uusi laskupohja](#)
[Tee uusi lasku](#)
[Näytä laskuraportit](#)

KÄYTTÄJÄHALLINTA

[Näytä kaikki käyttäjät](#)
[Luo uudet käyttäjätunnukset](#)

MUUT TOIMINNOT

[Palautelaatikko](#)
[Keskustelupalsta](#)
[Rekisteriseloste](#)

LUO UUSI LASKU

Valitse laskupohja:

Pankki:

Tilinumero:

Eräpäivä:

Maksuehto:

Kuva 26 Uuden laskun luonti.

Laskulle tulevat maksut valitaan laskupohjalle määritellyistä maksuista (kts. Kuva 27).

KOTKAN MELOJAT RY

JÄSENREKISTERI

Kirjautuneena: jonna [Kirjaudu ulos](#)

JÄSENREKISTERI

[Näytä kaikki jäsenet](#)
[Lisää uusi jäsen](#)
[Etsi jäseniä](#)

JÄSENKIRJEET

[Lisää jäsenkirje](#)
[Näytä kirjepohjat](#)

LASKUTUS

[Luo uusi laskupohja](#)
[Tee uusi lasku](#)
[Näytä laskuraportit](#)

KÄYTTÄJÄHALLINTA

[Näytä kaikki käyttäjät](#)
[Luo uudet käyttäjätunnukset](#)

MUUT TOIMINNOT

[Palautelaatikko](#)
[Keskustelupalsta](#)
[Rekisteriseloste](#)

LUO UUSI LASKU

Valitse laskutettavat maksut:

Maksut 2009

Jäsenmaksu:	10.00 €	<input type="checkbox"/>
Jäsenmaksu alle 16 v:	5.00 €	<input type="checkbox"/>
Kalustomaksu:	20.00 €	<input type="checkbox"/>
Säilytysmaksu kajakki:	23.00 €	<input type="checkbox"/>
Säilytysmaksu poolo:	0.00 €	<input type="checkbox"/>
Säilytysmaksu kaksikko:	0.00 €	<input type="checkbox"/>
Säilytysmaksu inkkari:	0.00 €	<input type="checkbox"/>
Säilytysmaksu Mansikkalahti:	0.00 €	<input type="checkbox"/>
Vajalla yöpyminen:	0.00 €	<input type="checkbox"/>
Alkeiskurssi:	0.00 €	<input type="checkbox"/>
Huomautusmaksu:	0.00 €	<input type="checkbox"/>


Kuva 27 Laskutettavien maksujen valinta.

Kuva 28 Laskun saajien valinta.

Laskun saajat listataan ja listalta voidaan vielä tarvittaessa poistaa laskunsaajia (kts. Kuva 28). Maksettavat maksut valikoituvat automaattisesti jäsenille sen perusteella mitä maksuja jäsentiedoissa kullekin jäsenelle on valittu. Esimerkiksi jäsenen, jolle on merkitty jäsenmaksu ja säilytysmaksu, laskuun ei tule kaluston käyttömaksua, vaikka se laskupohjassa olisikin valittuna. Laskutuksesta näytetään yhteenveto, jonka jälkeen laskut voidaan tulostaa. Laskutuksesta muodostuu automaattisesti raportti, jossa näkyvät jäsenten maksut, laskujen viitenumerot, eräpäivät ym. tiedot.

Jäsentietoihin tulee automaattisesti merkintä, että jäsen on ollut mukana jossain laskutuksessa ja jäsentietoihin voidaan merkitä laskut maksetuiksi. Maksetut laskut voidaan merkitä myös usealla jäsenelle yhtä aikaa, jottei jokaisen jäsenen jäsentietoja tarvitse ensin erikseen avata.

Alla olevassa kuvassa (kts. Kuva 29) on esimerkki jäsenrekisteristä tulostettavasta laskusta.

 KOTKAN MELOJAT RY		LASKU	
etunimi sukunimi katuosoite postinumero postitoimipaikka		Laskun pvm 1.4.2012	Laskun numero 1234567
		Maksuehto 14 pv	Eräpäivä 1.4.2012
		Viitteen 30009	Viivätykskorko 8%
		Laskutuskausi Ajalta: 1.1.-31.12.2012	

LASKUN ERITTELY

Jäsenmaksu	alle 16 v. 15€ / M uut 25€
Kalustomaksu	45€

HINNASTO

Jäsenmaksu	alle 16 v. 15€ / M uut 25€
Kalustomaksu	45€
Säilytysmaksu: koskarit ja poolot	10€
Säilytysmaksu: retkikajakit ja kaksikot	20€
Säilytysmaksu: inkkari	30€

KÄYTÄ VIITENUMEROA MAKSAESSASI LASKUA.

Saajan tilinumero FI83 1598 3006 1065 79		BIC NDEAFIHH	
Saaja Kotkan Melojat ry			
Maksajan nimi ja osoite Matti Mallikas Merikatu 8, 48100 Kotka			
Aikaispoitus		Viite 30009	
Tilin no		Eräpäivä 1.4.2012	Euro 25,00

Lasku lähetetään saajalle maksajarekisterin
 mukaisesti ja sen maksajan ilmoittaman tilinumeron
 perusteella.

Kuva 29 Esimerkki laskusta.

5.4.11 Jäsenkirjeet

Uudessa jäsenrekisterisovelluksessa on toiminto, jonka avulla jäsenille voidaan luoda ja lähettää jäsenkirjeitä. Jäsenkirjeiden luonti aloitetaan valitsemalla vastaanottajat (kts. Kuva 30). Vastaanottajat voidaan valita erilaisin hakuperustein. Tarvittaessa voidaan hakea esimerkiksi vain ne jäsenet, joilla on sähköpostiosoite merkittynä jäsentietoihin.

KOTKAN MELOJAT RY

JÄSENREKISTERI

Kirjautuneena: jonna [Kirjaudu ulos](#)

JÄSENREKISTERI

[Näytä kaikki jäsenet](#)
[Lisää uusi jäsen](#)
[Etsi jäseniä](#)

JÄSENKIRJEET

[Lisää jäsenkirje](#)
[Näytä kirjepohjat](#)

LASKUTUS

[Tee uusi lasku](#)
[Näytä laskuraportit](#)

KÄYTTÄJÄHALLINTA

[Näytä kaikki käyttäjät](#)
[Luo uudet käyttäjätunnukset](#)

MUUT TOIMINNOT

[Palautelaatikko](#)
[Keskustelupalsta](#)
[Rekisteriseloste](#)

JÄSENKIRJE

Valitse vastaanottajat:

---Valitse---

Hae vastaanottajat

Valitse kaikki Poista valinnat

Vastaanottajia kpl:

Kuva 30 Jäsenkirjeen luonti.

Jäsenkirjeen viesti kirjoitetaan valmiiseen pohjaan, jossa on yhdistyksen logo ja kirjeet voidaan tulostaa lähetettäväksi postitse tai ne voidaan lähettää suoraan jäsenten sähköposteihin.

5.5 Ei-toiminnalliset vaatimukset

Toiminnallisten vaatimusten lisäksi, ohjelmistotuotteet sisältävät joukon ei-toiminnallisia vaatimuksia, kuten laatu-, käytettävyys-, suorituskky ja tietoturva vaatimuksia. Jäsenrekisterisovelluksen ei-toiminnallisia vaatimuksia on kuvattu tarkemmin seuraavissa kappaleissa.

5.5.1 Laatuvaatimukset

Ohjelmistotuotteen laadulla tarkoitetaan yleisesti sovelluksen käyttökelpoisuutta. Määritelmä ei kuitenkaan kata suurtakaan osaa niistä piirteistä, joilla ohjelmistojen laadukkuutta voidaan mitata. Modernimpi laatuajattelu avartaa laadun määritelmää kattamaan määriteltyjen vaatimusten lisäksi myös muita abstraktimpeja ominaisuuksista. McConnellin mukaan laatu kattaa kaikki piirteet, joiden mielletään kuuluvan ohjelmistotuotteeseen – käytettävyys, tehokkuus, vakaus ja niin edelleen. (McConnell 2002:

127). Seuraavissa kappaleissa on kuvattu tarkemmin jäsenrekisterisovellukselle määritettyjä ei-toiminnallisia vaatimuksia. Osa ei-toiminnallisista vaatimuksista liittyy kiinteästi sovelluksen käytettävyyteen ja tietoturvaan sekä niiden asettamiin vaatimuksiin. Toiminnalliset ja ei-toiminnalliset vaatimukset muodostavat yhdessä vaatimukset, joista koostuu koko tuotteen laatu.

5.5.2 Käytettävyys

Suomen Standardisoimisliitto SFS ry:n (SFS ry 2011) mukaan käytettävyys on sitä kuinka helposti, oikein ja miellyttävästi käyttäjät pääsevät tavoitteisiinsa käyttäessään jotakin järjestelmää tai laitetta. Käytettävyys muodostuu useasta eri osa-alueesta kuten siitä miten hyvin käyttäjät suoriutuvat tehtävistään ja kuinka yksinkertaista ja tehokasta tuotteen, esimerkiksi sovelluksen, käyttö on.

Sovellusten ja järjestelmien käytettävyyttä voidaan parantaa huomioimalla ja toteuttamalla sitä jo suunnitteluvaiheessa. Käytettävyyden uranuurtajana pidetty Jakob Nielsen on luonut 10 heuristista sääntöä, joiden avulla käytettävyyttä voidaan parantaa:

1. Tuotteen tilan näkyvyys
2. Tuotteen ja tosielämän vastaavuus
3. Käyttäjän kontrolli ja vapaus
4. Yhteneväisyys ja standardit
5. Virheiden estäminen
6. Tunnistaminen mieluummin kuin muistaminen
7. Käytön joustavuus ja tehokkuus
8. Esteettinen ja minimalistinen design
9. Virhetilanteiden tunnistaminen, ilmoittaminen ja korjaaminen
10. Opastus ja ohjeistus (Nielsen 1990)

Nielsen on tehnyt myös listan pahimmasta web-suunnitteluvirheestä. Listaa päivittää edelleen ja se antaa hyvät peruslähtökohdat web-sovellusten suunnittelulle. Käytettävyyteen vaikuttavat perussäännöt eivät ole hankalia tai vaikeasti toteutettavia, mutta silti niitä rikotaan ja käytettävyyden suunnitelmallista toteutusta laiminlyödään jatkuvasti.

5.5.3 Helppokäyttöisyys, loogisuus ja opittavuus

Jäsenrekisterisovelluksen kaltaista ohjelmistoa ei ole tarkoitettu jokapäiväiseen käyttöön ja käyttäjät käyttävät sitä satunnaisesti. Erityisesti siksi sovelluksen on oltava helppokäyttöinen ja nopeasti opittava, etteivät käyttäjät turhaudu. Sovellusten tarkoituksena on helpottaa jonkin toiminnon suoritusta tai halutun päämäärän toteutumista. Käyttöliittymässä on käytettävä ”käyttäjän ymmärtämää kieltä” ja tuttuja sanoja, eikä esimerkiksi sovelluksen omaa sisäistä erikoissanastoa sanastoa.

Sovelluksen sivujen on oltava yhtenäiset niin ulkoasun kuin toiminnallisuudenkin osalta (esim. jos toiminto tehdään sivulla tietyllä tavalla, se tehdään samoin myös toisella sivulla tai jos sivulla on linkki etusivulle, sen on oltava samassa paikassa myös muilla sivuilla).

Käyttäjiä ei saa pakottaa muistamaan mitään, vaan kaikki tarpeellinen näytetään ko-koajan (esim. linkkilista). Käyttäjille on myös annettava riittävät visuaaliset vihjeet, eikä sovellusobjekteista saa poistaa valmiita visuaalisia vihjeitä esim. tyyli- tiedostojen avulla. Toisin sanoen tämä tarkoittaa sitä, että painikkeiden on näytettävä painikkeilta ja linkkien linkeiltä, eikä esimerkiksi painikkeita saa ohjelmoida linkeiksi.

5.5.4 Esteettömyys

Käyttäjää ei saa pakottaa tekemään mitään ylimääräistä tai lopputuloksen kannalta merkityksetöntä saadakseen haluamansa. Käyttäjä saa tehdä mitä haluaa ja milloin haluaa, eli käyttäjän on päästävä halutessaan jokaiseen toimintoon suoraan linkistä, tarvitsematta kulkea pitkää linkkipolkua löytääkseen haluamansa.

Käyttäjän on aina tiedettävä missä kohtaa sivustoa ollaan ja miten sieltä pääsee pois (esim. etusivulle). Lisäksi käyttäjän on tiedettävä koko ajan, mistä tarvittavat tiedot saa ja mistä saa apua, jos sitä tarvitaan.

5.5.5 Virhetilanteet

Hyvä sovellus estää käyttäjää tekemästä virheitä. Hyvin toteutettu sovellus tunnistaa mahdolliset ongelmatilanteet ja estää käyttäjää joutumasta pulaan. Virhetilanteita voidaan estää esim. huomioimalla seuraavia seikkoja:

- Estetään virheellisten syötteiden lähettäminen (esim. tarkistetaan, että sähköpostiosoitteessa on sähköpostiosoitteen tunnusmerkit, puhelinnumerokenttään voi kirjoittaa vain numeroita)
- Selkeä opastus “ongelmallisissa” tilanteissa (esim. ohjeistus ja selkeät, opastavat virheilmoitukset)

Virheen sattuessa, käyttäjälle on kerrottava selkeästi: mitä tapahtui, miksi, miten sen voi korjata ja kuinka se voidaan välttää jatkossa. Virheilmoitusten on oltava selkeitä ja yhtenäisiä kautta koko sovelluksen. Jos käyttäjä tekee virheellisen valinnan, hänen on päästävä helposti myös perumaan ja korjaamaan se.

5.5.6 Suorituskyky ja vasteajat

Käyttäjät vierailevat web-sivuilla yleensä hyvin vähän aikaa. Jos haluttua ei saada heti, käyttäjä lähtee yleensä pois ja etsii jotain parempaa. Sivustolla on n. 7 sekuntia aikaa vakuuttaa käyttäjä, eli sivuston latautumisen on oltava nopeaa ja sujuvaa. Sivujen latautumisaika tulee olla korkeintaan n. 1 sekunti, jotta käyttäjä kokisi pystyvänsä liikumaan sivustolla esteettömästi. Riippuu paljon toiminnosta, kauanko se saa kestää. Esimerkiksi painikkeen painamisesta vasteen on tultava heti, mutta esimerkiksi tiedoston lataaminen sivulta saa kestää kauemmin.

Sovelluksen suorituskykyyn vaikuttavat useat eri laite- ja ohjelmistokomponentit. Nopeuteen vaikuttavat monet asiat, jotka kaikki eivät ole pelkästään koodiriippuvaisia (esim. kuvat, grafiikat jne.). Jäsenrekisterisovelluksessa minkään toiminnon ei pitäisi kestää niin kauan, että kestosta pitäisi ilmoittaa käyttäjälle esim. aikapalkeilla. Sivuja saadaan nopeutettua mm. jättämällä pois kaikki turha (ei raskaita upotettuja objekteja tms.) ja käyttämällä sivuston tila mieluummin hyötykäyttöön. Jäsenrekisterisovellukselle asetettiin lisäksi vaatimukseksi, että sen on tuettava riittävää määrää yhtäaikaista käyttäjiä.

5.5.7 Muita ei-toiminnallisia vaatimuksia

Jäsenrekisterin suunnitteluvaiheessa määriteltiin ja suunniteltiin myös seuraavia ei-toiminnallisia vaatimuksia:

- Selainikkunat

- Palvelu ei saa avata turhia selainikkunoita, vaan se pitää pystyä toteuttamaan yhdessä ikkunassa
- Sovelluksen skaalautuminen
 - Sovellusikkunoiden on skaalauduttava eri kokoisille näytöille.
- Palvelun palauttaminen
 - Palvelu ja sen sisältämät tiedot on oltava palautettavissa laitevian tai muun vakavan tilanteen jälkeen nopeasti siihen tilaan, jossa se oli ennen tapahtumaa.

5.6 Tietoturva

Tietoturva voitaisiin lukea osaksi ei-toiminnallisia vaatimuksia (kts. 5.5 Ei-toiminnalliset vaatimukset s. 59), mutta laajuutensa vuoksi sitä käsitellään tässä omalla osionaan. Tietoturvalla tarkoitetaan yleisesti tietojen ja palvelujen suojaamista normaali- ja poikkeustilanteissa ja se rakentuu tiedon luottamuksellisuuden, eheyden ja saatavuuden turvaamisesta. Luottamuksellisuudella tarkoitetaan sitä, että tiedot ja palvelut ovat vain niihin oikeutettujen henkilöiden saatavissa. Eheydellä varmistetaan, ettei tieto ole muuttunut tai tuhoutunut ja saatavuudella taataan tietojen esteetön käyttö niille, jotka ovat niihin oikeutettuja.

Tietoturvallisuuden toteuttamiseksi tehtävät toimenpiteet jaetaan yleensä kahdeksaan osa-alueeseen:

- hallinnollinen ja organisatorinen tietoturvallisuus
- henkilöstöturvallisuus
- fyysinen turvallisuus
- tietoliikenneturvallisuus
- laitteistoturvallisuus
- ohjelmistoturvallisuus
- tietoaineistoturvallisuus
- käyttöturvallisuus (Rousku 2003).

Tietoturvalla ei tarkoiteta siis pelkästään sovellusten tietoturvaa, vaan myös tietojen fyysistä turvaamista luvattomalta käytöltä. Seuraavissa kappaleissa on kuvattu erilaisia sovelluksiin liittyviä tietoturvauhkia ja miten niitä huomioidaan jäsenrekisterisovelluksessa.

5.6.1 Identiteettivarkaudet

Laajat identiteettivarkaudet ovat lisääntymässä niin Suomessa kuin muuallakin maailmassa. USA:n kuluttajavalituslautakunta FTC (Federal Trade Commission) julkaisee vuosittain listan yleisimmistä kuluttajavalituksista ja vuonna 2011 identiteettivarkaudet olivat listan kärjessä kahdettatoista kertaa peräkkäin. (FTC 2012) Iso-Britanniassa identiteettivarkauksien määrä kasvoi 66% vuosina 2006-2007. (Financial Risk Management News And Analysis 2012).

Identiteettivarkaus ei ole rikoslain termi eikä identiteetin rakentaminen ole Suomessa rikollista ellei tietoja kerätä esim. murtautumalla johonkin tietojärjestelmään tai jos toisen henkilötietoja käytetään hyötymistarkoituksessa esimerkiksi petoksissa tai asiakirjaväärennöksissä. Yhä useammin identiteettivarkauden uhrien tiedot saadaan internetistä vakoiluohjelmien tai huijausviestien avulla tai murtautumalla sähköisiin rekistereihin.

5.6.2 Tietovuodot ja tietomurrot

Identiteettivarkauksien myötä viime vuosina myös muiden tietovuotojen ja -murtojen määrä on kasvanut. Suomen tähänastiseksi suurimmaksi tietomurroksi nimitetty tapaus tuli ilmi 5.11.2011, kun 16 000 suomalaisen henkilötiedot vuodettiin internetiin. (Viestintävirasto, CERT-FI 2011) Viikkoa myöhemmin internetissä julkaistiin 500 000 suomalaisen sähköpostiosoitteista (MikroPC 2011) koostuva lista ja sen jälkimainingeissa verkossa alkoi kiertää 14 600 salasanan lista, joista ainakin osa on todistettu aidoiksi (MikroPC 2011). 6.6.2012 6,5 miljoonaa, todennäköisesti LinkedIn -verkkoyhteisöpalvelusta varastettua salasananatiivistettä, julkaistiin venäläisellä hakkerisivustolla. Ainakin 250 000 salasananatiivistettä oli jo pystytty avaamaan (Viestintävirasto, CERT-FI 2012).

5.6.3 Salasanojen turvallisuus

Salasanojen turvallisuuteen liittyy useita seikkoja. Osa turvallisuuteen vaikuttavista seikoista on salasanojen luontitapa, tallennustapa tietokantaan, käytetty salaustai tiivistystekniikka, suolaus, sekä salasanojen käsittely esimerkiksi verkkosovelluksissa. Yhä enemmän salasanojen turvallisuuteen vaikuttaa kuitenkin käyttäjien oma toiminta. Kaikkiin palveluihin tarvitaan nykyisin salasanoja ja käyttäjätunnuksia, joten nii-

den muistaminen ja käsittely on hankalaa. Tästä syystä käyttäjät vaarantavat tahtomattaan monien järjestelmien tietoturvan luomalla helposti arvattavia ja murrettavia salasanoja, käyttämällä samaa salasanaa kaikissa palveluissa ja säilyttämällä salasanoja sähköposteissaan.

Useissa palveluissa tunnukset lähetetään käyttäjän sähköpostiin ja niissä on toiminto, jolla unohtunut salasana pystytään lähettämään uudelleen sähköpostiin. Sähköpostia käytetäänkin usein varastona salasanoille, eikä palvelujen lähettämiä viestejä poisteta. Siksi murrettua sähköpostilaatikkoa voidaan pitää ”yleisavaimena”, jonka sisältämällä tunnuksilla päästään useisiin palveluihin. Sähköpostin lisäksi tunnus-salasanapareja säilytetään huolimattomasti paperille kirjoitettuna, mitä voitaisiin verrata pankkikortin numeron säilyttämiseen lompakossa kortin vieressä.

Alla olevissa taulukoissa on esitetty salasanojen murtonopeuksia eri tapauksissa. Salasanat on murrettu brute-force -menetelmää hyväksi käyttäen. Ensimmäisessä taulukossa (Taulukko 3) esitetään, montako salasanaa taulukossa oleva prosessori voi tarkastaa sekunnissa. Toisessa taulukossa (Taulukko 4) on kuvattu aika, kuinka kauan Pentium III -koneelta kului eri salasanojen murtamiseen ja viimeisestä taulukosta (Taulukko 5) nähdään, millainen vaikutus on, jos perusaakkosten lisäksi salasanoissa käytetään merkkejä ja numeroita.

Taulukko 3 Taulukossa näytetään, montako salasanaa prosessori voi tarkastaa sekunnin aikana.

Kone	salasanoja/sekunti
Pentium I 133 MHz	17 600
Pentium II 266 MHz	48 000
Pentium III 933 MHz	171 700

Taulukko 4 Pentium III -koneelta kulunut aika eri salasanojen murtamiseen.

Salasana	Aika	Kokeiltuja salasanoja
mu	0 s	360
mui	0,1 s	9 344
muik	1,0 s	159 612
muikk	38 s	6 316 166
muikku	16 min 40 s	164 220 312
muikkuj	13 h 30 min 7 s	4 269 728 096

Taulukko 5 Salasanojen monimutkaisuuden vaikutus salasanan murtonopeuteen.

Salasana	Mitä kokeillaan	Kulunut aika
muikku	pienet	1 min 40 s
MuikKu	pienet, isot	8 h 58 min
Mui789	pienet, isot, numerot	1 vrk 13 h 43 min
Mui-89	pienet, isot, numerot, erikoismerkit	19 vrk 13 h 27 min

(Järvinen 2002)

Yleensä salasana tallennetaan tietokantaan salattuina tai tiivistettyinä jollakin tiivistysalgoritmeilla. Ennen tiivistystä salasanoihin lisätään suolaus, eli satunnainen bittijono, joka vaikeuttaa sanakirja- ja rainbow table -hyökkäyksiä.

Jäsenrekisterin toiminnot suojataan käyttäjätunnus-salasana -pareilla, joiden avulla todennetaan käyttäjät, joilla on pääsy rekisterin tietoihin. Käyttäjätunnuksia on kahta tasoa: jäsentason tunnuksia ja ylläpitotason tunnuksia. Jäsentason tunnuksilla on pääsy ainoastaan jäsenen omiin tietoihin, kun taas ylläpitotason tunnuksilla voidaan nähdä kaikki rekisterin tiedot. Ylläpitotason tunnuksia jaetaan vain Kotkan Melojat ry:n hallituksen jäsenille. Salasanan vähimmäispituudeksi on määritetty 8 merkkiä, eikä maksimipituutta ole säädetty. Salasana tallennetaan tietokantaan suojattuina ja suolattuina, eli ne eivät ole tietokannassa selkokielisiä.

5.6.4 Istuntokaappaukset

Palvelimella toimivaan web-palveluun voidaan yrittää hyökätä istuntokaappauksen avulla. Istunnot (sessions) ovat esimerkiksi PHP-muuttujia, joiden sisältämät arvot säilyvät muistissa web-sivulta toiselle siirryttäessä koko istunnon ajan. Tyypillinen esimerkki istuntojen käytöstä on web-palveluun kirjautuminen. Istunnot perustuvat

evästeisiin (cookies), joihin tiedot tallennetaan. Evästeet voivat olla joko istuntokohtaisia (session cookies) tai pysyviä (persistent cookie). Istuntokohtaiset evästeet vanhenevat, kun käyttäjä sulkee istunnon ja pysyviä evästeitä säilytetään tietyn aikaa tai kun käyttäjä tuhoaa ne. Uusi istunto luodaan, kun käyttäjä kirjautuu sisälle palveluun. Käyttäjälle muodostetaan väliaikainen uniikki id-tunnus, joka tallennetaan evästeeseen PHPSESSID. Palvelimelle luodaan tiedosto, jonka nimeksi annetaan käyttäjän väliaikainen id ja tiedostoon tallennetaan istunnon aikana käytettävät istuntomuuttujat. Evästeet on mahdollista kytkeä pois käytöstä selaimessa.

Istuntokaappaus toteutetaan käyttämällä toisen käyttäjän istuntotunnistetta ja huijaamalla näin palvelua pitämään hyökkääjää luvallisena käyttäjänä. Istuntotunniste voidaan saada selville mm. arvaamalla tai kuuntelemalla salaamattomana lähetettyä viestiliikennettä esimerkiksi langattomassa verkossa. URL-osoitteen mukana siirrettävät istuntotunnistukset voidaan kaapata myös vihamielisen linkin tai koodin avulla. Jotta web-palveluun ei voitaisi liittää vihamielistä koodia, tulisi käyttäjien antamat syötteet puhdistaa kunnolla, eikä käyttäjille saisi koskaan antaa mahdollisuutta liittää html- tai skriptikoodia suoraan sivustolle. Istuntokaappauksilta voidaan suojautua esimerkiksi salamaalla tietoliikenne tai vaihtamalla istuntotunnusta aina palveluun kirjaututtaessa.

Jäsenrekisterin kirjautuminen on toteutettu istuntojen avulla. Jokaisella kirjautumiskerralla luodaan uusi istuntotunnus, joka tiivistetään eli se ei ole saatavilla selkokielenä. Sovellus käyttää PHP:n istunnonhallintaa, eikä tallenna käyttäjätietoja evästeisiin. Käyttäjän kirjautuessa ulos palvelusta, istunto tuhotaan.

5.7 Tietokannan suunnittelu ja normalisointi

Tietokannan suunnittelua tehtiin samanaikaisesti toiminnallisten määrittelyjen kanssa. Toiminnalliset määrittelyt ja vaatimukset vaikuttavat paljon tietokannan rakenteeseen ja sen sisältämiin tietoihin. Tietokannan suunnittelu aloitettiin listaamalla tarvittavat tiedot ja täydentämällä listaa sitä mukaa, kun uusia tietoja ilmestyi toiminnallisen määrittelyn edetessä. Tietokannan suunnittelussa käytettiin normalisointimenetelmää.

Relaatiotietokantamallin kehittäjä Edgar F. Coddin suunnitteli normalisointimallin, jonka avulla tietokanta saadaan järjestettyä niin, ettei se sisällä päällekkäisyyksiä. Normalisoinnilla saadaan myös lisättyä tietokantarakenteen selkeyttä ja yhtenäisyyttä, se auttaa tietokannan ylläpitoa, vähentää tilantarvetta ja helpottaa tietokannan laajen-

nettavuutta myöhemmin. Coddin alkuperäisessä mallissa on kuvattu kolme normaalimuotoa, mutta myöhemmin malliin on lisätty vielä kaksi normaalimuotoa, mutta yleisesti tietokantaa pidetään kuitenkin normalisoituna, jos se täyttää normalisoinnin ehdot neljänteen normaalimuotoon asti.

Normalisoinnin tuloksena tietokantaa kehitetään asteittain siten, että sen rakenne tarkentuu koko ajan. Tietokannan normalisointi aloitetaan siis tietokannan suunnitteluvaiheessa. Tietokannan suunnittelu aloitetaan listaamalla kaikki tietokantaan tarvittavat tiedot ja jakamalla ne tauluihin. Tauluja aletaan normalisoida, minkä tuloksena voidaan huomata, jos taulun rakennetta kannattaa muuttaa tai tiedot on jaettava toisin.

Microsoftin (Microsoft 2012) tuotetukisivustolla on kuvattu kolme ensimmäistä normalisointivaihetta seuraavasti:

Ensimmäinen normaalimuoto

- Poista yksittäisissä taulukoissa toistuvat ryhmät.
- Luo erillinen taulukko kullekin liittyvien tietojen joukolle.
- Merkitse kukin liittyvien tietojen joukko perusavaimen avulla (Microsoft 2012).

Ensimmäisessä normalisointivaiheessa on siis tarkoituksena määrittää jokainen tietokannan kenttä ydintiedoksi siten, että tieto pilkotaan sen pienimpään muotoon. Esimerkiksi etunimeä ja sukunimeä ei saa laittaa samaan kenttään, sillä se vaikeuttaisi mm. tietojen lajittelua, jos jossain kentissä olisi ensin etunimi ja toisissa sukunimi. Lisäksi ensimmäisessä vaiheessa tiedot lajitellaan omiin ryhmiinsä siten, että yhteen ryhmään tulee vain samaan ryhmään liittyviä tietoja. Esimerkiksi henkilötietoja sisältävään tauluun ei laiteta laskuihin liittyviä tietoja.

Toinen normaalimuoto

- Luo erilliset taulukot useita tietueita koskeville arvojen joukoille.
- Muodosta taulukoiden välinen suhde viiteavaimen avulla. (Microsoft 2012)

Toisessa normalisointikohdassa jokaiseen edellä luotuun tauluun uniikki viiteavain, eli kenttä, jonka avulla tietueet tunnustetaan yksilöllisesti.

Kolmas normaalimuoto

- Poista kentät, jotka eivät ole riippuvaisia avaimesta. (Microsoft 2012)

Kolmannessa normalisointivaiheessa edellytetään, että jokaisen kentän on oltava keskenään riippumaton, eli taulun sisältö ei saa perustua toisen taulun kenttään.

Kahdesta viimeisestä jälkeensä lisätystä normalisointivaiheesta Microsoftin (Microsoft 2012) tukisivustolla mainitaan ainoastaan seuraavaa:

Neljäs normaalimuoto (Boyce-Coddin normaalimuoto eli BCNF) ja viides normaalimuoto ovat olemassa, mutta niitä sovelletaan käytännön suunnittelussa vain harvoin. Näiden sääntöjen jättäminen huomiotta saat-
taa antaa tulokseksi muun kuin täydellisen tietokantarakenteen, mutta sen ei pitäisi vaikuttaa tietokannan toimintaan.

5.8 Tekninen määrittely

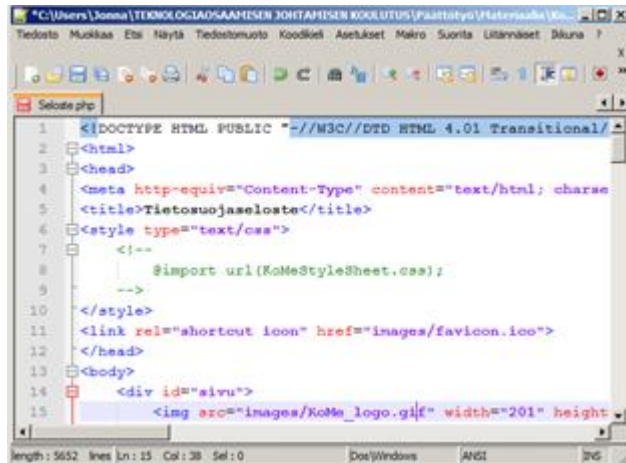
Tekninen määrittely kuvaa jäsenrekisterisovelluksen teknisiä ratkaisuja kuten alustaa käytettyjä ohjelmointikieliä, tekniikoita sekä työkaluja.

5.8.1 Webhotelli

Kotkan Melojat ry:n jäsenrekisterin palveluntarjoajana toimii helsinkiläinen Webhotelli.fi, joka on erikoistunut webhotellipalvelujen tarjoamiseen kuluttajille, yhteisöille ja pienille yrityksille. Yhdistyksen kotisivut toimivat aiemmin toisella palveluntarjoajalla, mutta tarvittavien toimintojen ja hinnan vuoksi palvelualusta vaihdettiin. Webhotelli.fi tarjoaa neljää erilaista palvelupakettia, joista palvelujen puitteissa jäsenrekisterille katsottiin riittävän toiseksi pienin. Palvelupakettiin kuuluu mm. 2000 MB levytilaa, 20 GB/kk liikennöintirajan, 10 sähköpostilaatikkaa ja MySQL-tietokannat. Ohjelmisto on toteutettu pääosin PHP-ohjelmointikielellä sekä HTML-merkkauksielellä. Ohjelmistoympäristö on yleinen kokoonpano verkkopalveluja toteutettaessa.

5.8.2 Notepad++

Jäsenrekisterin koodaukseen käytettiin vapaaseen lähdekoodiin perustuvaa Notepad++ -koodieditoria (Kuva 31). Notepad++ -sovellus toimii Windows -ympäristössä ja sen pohjana käytetään lähdekoodin editointiin tarkoitettua scintilla -komponenttia.



Kuva 31 Näyttökaappaus Notepad++ -ohjelmasta.

Notepad++ tukee useita eri merkintä-, ohjelmointi- ja scriptikieliä, kuten jäsenrekisterin koodauksessa käytettyjä PHP:tä, HTML:ää, CSS:ää sekä JavaScriptiä. Notepad++ helpottaa koodausta korostamalla käytetyn kielen syntakseja eli värjäämällä kirjoitettavan koodin. Se mahdollistaa myös kirjoitettavien sanojen täydennyksen automaattisesti sekä useiden koodisivujen yhtäaikaisen auki olemisen.

Notepad++ on toteutettu GPL-lisenssin (General Public License) pohjalle. GPL on GNU (GNU's Not Unix) -hankkeen yleisin lisenssi, jota käytetään vapaiden ohjelmistojen julkaisuun. GNU hanketta hallinnoi FSF (Free Software Foundation). GPL -lisenssi antaa kaikille oikeuden käyttää, kopioida, muuttaa ja edelleen jakaa ohjelmia sekä ohjelmien lähdekoodia. Lisenssin tarkoituksena on taata, että samat oikeudet säilyvät muunnelluissa sovelluksissa, eli jos muunnettua koodia levitetään edelleen, on sen lähdekoodin oltava julkinen, eikä ohjelman tai koodin käytölle saa asettaa rajoituksia. Notepad++ -sovelluksen virallisilla sivuilla ilmoitetaan, että Notepad++ pyrkii vähentämään hiilidioksidipäästöjä optimoimalla mahdollisimman paljon rutiineja. Näin ollen tietokone käyttää vähemmän prosessoritehoa ja säästää näin ympäristöä.

5.8.3 PHP

Jäsenrekisterin toiminnallisuuden ohjelmoinnissa käytettiin lähinnä The PHP Groupin kehittämää PHP-ohjelmointikieltä. PHP-kieli on kehitetty erityisesti dynaamisten web-sivustojen luontiin ja kuten yleisestikin, koodi on upotettu HTML-kielen sisään. PHP on palvelin pohjainen kieli, eli koodi käännetään palvelimella ennen sivun lähetyksen selaimelle. Tästä syystä kieli ei vaadi selaintukea. Jäsenrekisterin käytössä oleva PHP-versio on versio 5.

5.8.4 JavaScript

Joihinkin jäsenrekisterissä tehtäviin tarkistuksiin ja virheilmoitusten tulostukseen dialogi -ikkunoihin on yksinkertaisuuden vuoksi käytetty JavaScripteja. Päinvastoin, kun PHP, JavaScript on yksinkertainen selaimessa suoritettava koodi. Kaikki nykyaikaiset web-selaimet tukevat JavaScriptiä, mutta koodin suoritus voidaan kuitenkin estää selaimen asetuksista, joten sillä ei voida toteuttaa tärkeitä toimintoja.

5.8.5 MySQL

Jäsenrekisterin tietokantana toimii PHP:n kanssa yleisimmin käytettävä tietokantatyyppi MySQL. Tietokantakyselyt toteutetaan SQL-kielillä.

5.8.6 CSS

CSS (Cascading Style Sheets) on web-sivuille kehitetty tyyli tiedostoformaatti, jonka avulla html-sivujen ulkoasua voidaan muokata halutuksi. CSS:n avulla kaikki sovellukseen kuuluvat sivut saavat yhtenäisen ulkoasun ja graafisen ilmeen, jonka muuttaminen vaatii ainoastaan yhden tiedoston muokkaamista. CSS-tiedostossa määritellyt tyyliohjeet ovat ehdollisia, eli selaimet, jotka eivät voi toistaa niitä, voivat korvata ne muilla tyyliohjeilla.

5.8.7 Merkistökoodaus

Merkistökoodauksella muutetaan tekstissä käytettävät merkit tietokoneen ymmärtämiseksi numeroiksi, eli merkistökoodauksella määritellään mitä merkkejä tekstissä voi-

daan käyttää. Yleisimmin web-ympäristöissä käytetään UTF-8 ja ISO-8859-1 -standardeja.

Jäsenrekisterisovelluksessa merkistökoodauksena käytetään ISO-8859-1 eli Latin 1 -standardia, joka on luotu erityisesti silmällä pitäen länsieurooppalaisia kieliä ja niissä käytettäviä merkistöjä. ISO-8859-1 sisältää 256 8-bittistä merkkiä ja sitä voidaan kutsua 7-bittisen ASCII-merkistön laajennukseksi. ASCII-merkistön ensimmäiset 128 merkkiä käsittävät englanninkielessä käytettävät aakkoset, numerot sekä yleisimmät välimerkit. ISO-8859-1 -standardi määrittelee lisäksi länsieurooppalaisissa kielissä käytettäviä merkkejä, kuten ä- ja ö-kirjaimet. ISO-5589-1-merkistökoodauksen seuraajaksi on kehitetty ISO-8859-15 (Latin 9), joka sisältää esim. ISO-5589-1:stä puuttuvan euron merkin (€). Standardi ei kuitenkaan ole vielä syrjäyttänyt edeltäjänsä.

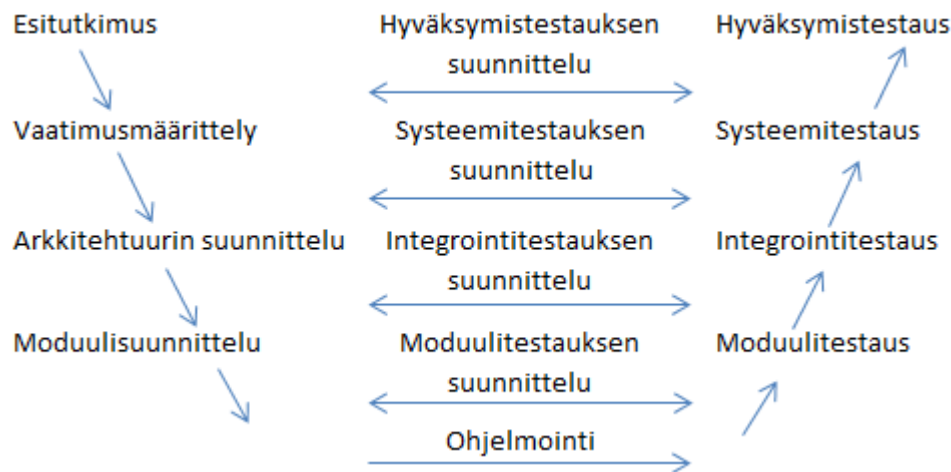
Jos merkistökoodaus ei tue käytettäviä merkkejä, ne näkyvät selaimessa väärin. Merkistöistä puuttuvat merkit voidaan korvata niin kutsutuilla entiteettimerkeillä. Entiteettimerkkejä käytetään merkintäkielissä, kuten html-kielessä, korvaamaan merkkejä, joille merkintäkielessä on varattu erityismerkitys tai ne eivät sisälly käytettävään merkistöön. Entiteettimerkit aloitetaan & -merkillä ja ne päättyvät puolipisteeseen (;). Esimerkiksi euron merkin entiteetti on € ja se näkyy selaimessa €-merkkinä.

5.9 Testaussuunnitelma

Ohjelmatestauksen tavoitteena on parantaa ohjelman laatua vähentämällä sen virheiden määrää. Testauksen päämääränä ei voida pitää virheetöntä ohjelmaa, vaan ohjelmaa, jossa on mahdollisimman vähän virheitä.

Mitä aikaisemmassa ohjelmistokehityksen vaiheessa testaus aloitetaan, sitä paremmin se toimii ja sitä enemmän virheitä löytyy ja korjataan. Ohjelmistotuotannon myöhemmissä vaiheissa löytyneiden virheiden korjaus on hankalaa, koska monimutkaisen lähdekoodin muutokset saattavat helposti aiheuttaa uusia virheitä ja vaikuttaa myös jo toimiviin ominaisuuksiin. Lisäksi myöhemmissä vaiheissa ilmenneiden virheiden korjaaminen on aina kalliimpaa kuin, jos ne löydetäisiin aiemmin.

Alla olevassa kuvassa (Kuva 32) on perinteinen V-malli, joka kuvaa testauksen suhdetta muuhun ohjelmistoprojektiin. Kuvasta nähdään, että testaus- ja ohjelmistoprosessit etenevät tiiviisti rinnakkain koko ohjelmistoprojektin ajan.



Kuva 32 Mukailtu kuva perinteisestä V-mallista, joka kuvaa testauksen suhdetta muuhun ohjelmistoprojektiin.

Ohjelman testaus aloitetaan jo siinä vaiheessa kun ohjelman vaatimusmäärittelyä tehdään. Verifioinnilla eli todentamisella tarkoitetaan ohjelman toimintojen testausta juuri määrittelyihin nähden. Vaatimusmäärittelyä verifioidaan asiakkaan kanssa ja pyritään selvittämään vastaavatko määritellyt toiminnot ja ominaisuudet ohjelmalle asetetuja tarpeita. Määrittelyjä muokataan niin kauan, että asiakas on ne hyväksynyt. Testausta vaatimusmäärittelyä vastaan tehdään, poistamaan väärinymmärryksistä aiheutuvia virheitä tai puutteita.

Ohjelman testausta suoritetaan koko sovelluskehityksen ajan. Koodin testausta tehdään ohjelmoinnin yhteydessä niin sanottuna lasilaatikkotestauksena ja valmiita toimintoja testataan funktionaalisen testauksen avulla. Kun ohjelma on valmis asiakas suorittaa hyväksymistestauksen, jossa tutkitaan määrittelyjen ja tarpeiden vastaavuutta. Hyväksymistestaus suoritetaan sekä määrittelyjen yhteydessä että valmiille ohjelmalle. Testaus voidaan lopettaa, kun kaikki löydetyt virheet on korjattu, sovellus on testattu uudelleen, eikä uusia virheitä enää löydy.

Jäsenrekisterin testauksessa päätettiin käyttää lisäksi betatestausta ennen sovelluksen laajempaa käyttöönottoa. Betatestauksella tarkoitetaan valmiin tai lähes valmiin ohjelman antamista potentiaaliselle käyttäjryhmälle koekäyttöön. Testikäyttäjät raportoivat ongelmista ja virheistä sovelluksen toteuttajalle, joka korjaa ne.

5.9.1 Lasilaatikkotestaus

Lasilaatikkotestausta tehdään samaan aikaan ohjelman koodin kirjoittamisen kanssa. Lasilaatikkotestauksessa testaajalla on käytettävissään ohjelman lähdekoodi ja siinä käydään läpi ohjelman sisäistä rakennetta, kuten lauseiden ja haarojen kattavuutta ja ohjelman sisäistä logiikkaa.

Koska lähdekoodi on koko ajan nähtävillä, testitapauksien valinta on helppoa. Koodista nähdään suoraan mitkä syötteet ovat sallittuja, millaisten syötteiden kanssa on mahdollisesti ongelmia, mitä syötteitä kannattaa valita ja miten ohjelman tulisi niihin reagoida. Esimerkiksi numeerisia arvoja testattaessa valitaan suurin ja suurin sallittu arvo sekä suurinta sallittua arvoa suurempi ja pienintä pienempi arvo.

5.9.2 Funktionaalinen testaus eli mustalaatikkotestaus

Funktionaalisessa testauksessa eli mustalaatikkotestauksessa testaaja ei näe ohjelman lähdekoodia eikä rakennetta, vaan testaus suoritetaan ainoastaan ohjelman määrittelyjen perusteella. Ohjelmaa testataan funktio tai toiminto kerrallaan. Toimintoja verrataan määrittelyyn joten myös määrittelyn ja ohjelman välistä yhdenmukaisuutta testataan. Funktionaalisella testauksella testataan siis valmista tai lähes valmista ohjelmaa ja sen määrittelyä. Funktionaalisessa testauksessa ohjelmalle annetaan jokin syöte, jonka ohjelma käsittelee ja palauttaa jonkin arvon. Palautusarvoa verrataan etukäteen tiedossa olevaan arvoon, siis arvoon johon funktion pitäisi määrittelyjen mukaan päätyä.

Funktionaaliseen testaukseen ja sen testiaineiston valintaan on kehitetty tiettyjä metodeja, jotta testattava aineisto saataisiin mahdollisimman pieneksi, mutta kuitenkin mahdollisimman kattavaksi. Koska metodit eivät yleensä yksinään ole kattavia, niitä yhdistellä yleensä toisiinsa sopiviksi kokonaisuuksikseen. Funktionaalisen testauksen apuna voidaan käyttää esimerkiksi seuraavia metodeja:

- Testiaineiston jako ekvivalenssiluokkiin
 - o Määrittelyn perusteella valitaan arvoja, jotka ovat ohjelman syöteinä sallittuja ja ei-sallittuja. Molemmista luokista valitaan mahdollisimman kattava testiaineisto jota testataan ohjelmalla.
- Reuna-arvoanalyysi

- Reuna-arvoanalyysissä pyritään valitsemaan testattavat syötteet niin, että ne ovat mahdollisimman suuria tai pieniä tai mahdollisimman pitkiä tai lyhyitä. Analyysi perustuu olettamukseen, että ohjelma epäonnistuu joillakin reuna-arvoilla.
- Virheen arvaus
 - Virheen arvausmenetelmässä testaajalla on oltava jonkinlainen mielikuva syötteistä, jotka saattavat aiheuttaa ongelmia (esimerkiksi nollajako), joten ne valitaan testiarvoiksi.

5.9.3 Web-selaimet palvelun testauksessa

Nykyisten ja tulevien www-standardien tuen testaamiseen selaimissa käytetään WaSP:n (The Web Standard Project) kehittämää Acid -testiä. Uusin, Acid3 -testi, julkaistiin 2008, eikä yksikään siihen mennessä olemassa oleva tai kehitteillä oleva www-selain suoriutunut testistä. Testiä kritisoitiin ja sitä jouduttiin muuttamaan, jonka jälkeen kaikki nykyisin käytössä olevat selaimet selviytyivät siitä.

IE-selain (Internet Explorer) on ollut kautta aikojen tunnettu www-standardija kunniottamattomista ratkaisuksistaan, joka vaikeuttaa www-sivujen tekoa. Validi www-koodi näkyy yleisesti ottaen oikein suurimmalla osalla selaimia, mutta IE:n käyttämät ratkaisut vaativat yleensä lisätestausta. IE:n kehityksessä on kuitenkin viime aikoina kiinnitetty enemmän huomiota www-standardien tukeen.

Alla olevassa kuvassa on W3C:n julkaisema tilasto eri selainten käytöstä. Eniten käytettyjä selaimia ovat Internet Explorer, Firefox ja Google Chrome, joka syrjäytti Firefoxin suosituimpana selaimena 2012. IE. Www-sivujen testauksessa on siis otettava huomioon ainakin eri selaimet ja vähintään sivut tulisi testata kaikilla yleisimmillä selaimilla ja niiden eri versioilla.

2012	<u>Internet Explorer</u>	<u>Firefox</u>	<u>Chrome</u>	<u>Safari</u>	<u>Opera</u>
September	16.4 %	32.2 %	44.1 %	4.2 %	2.1 %
August	16.2 %	32.8 %	43.7 %	4.0 %	2.2 %
July	16.3 %	33.7 %	42.9 %	3.9 %	2.1 %
June	16.7 %	34.4 %	41.7 %	4.1 %	2.2 %
May	18.1 %	35.2 %	39.3 %	4.3 %	2.2 %
April	18.3 %	35.8 %	38.3 %	4.5 %	2.3 %
March	18.9 %	36.3 %	37.3 %	4.4 %	2.3 %
February	19.5 %	36.6 %	36.3 %	4.5 %	2.3 %
January	20.1 %	37.1 %	35.3 %	4.3 %	2.4 %
2011	<u>Internet Explorer</u>	<u>Firefox</u>	<u>Chrome</u>	<u>Safari</u>	<u>Opera</u>
December	20.2 %	37.7 %	34.6 %	4.2 %	2.5 %
November	21.2 %	38.1 %	33.4 %	4.2 %	2.4 %
October	21.7 %	38.7 %	32.3 %	4.2 %	2.4 %
September	22.9 %	39.7 %	30.5 %	4.0 %	2.2 %
August	22.4 %	40.6 %	30.3 %	3.8 %	2.3 %
July	22.0 %	42.0 %	29.4 %	3.6 %	2.4 %
June	23.2 %	42.2 %	27.9 %	3.7 %	2.4 %
May	24.9 %	42.4 %	25.9 %	4.0 %	2.4 %
April	24.3 %	42.9 %	25.6 %	4.1 %	2.6 %
March	25.8 %	42.2 %	25.0 %	4.0 %	2.5 %
February	26.5 %	42.4 %	24.1 %	4.1 %	2.5 %
January	26.6 %	42.8 %	23.8 %	4.0 %	2.5 %

Kuva 33 Tilastot eniten käytetyistä selaimista vuosina 2011-2012 (W3C 2012).

5.10 Riskienhallinta ohjelmistoprojekteissa

Ohjelmistotuotantoon liittyy useita riskejä ja riskien hallinta voi olla vaikeaa, koska tuotteina ohjelmistot ovat luonnostaan monimutkaisia. Tutkimustalo Standish Groupin mukaan kaikista vuonna 2009 toteutetuista tietojärjestelmähankkeista vain 32 % onnistui (Tietoviikko 2010). Ohjelmistot toteutetaan usein projektiluonteisesti. Pohjosen (2007: 80) mukaan riskienhallinta on suunnittelemattoman hallintaa. Eräs tapa ohjelmistoprojektin riskien hallintaan on tunnistaa ja arvioida riskit, etsiä mahdollisia ratkaisuja niihin ja toteuttaa ratkaisut riskien minimoimiseksi tai poistamiseksi kokonaan.

Ohjelmistoprojektien riskien tunnistuksessa kannattaa miettiä, miksi ohjelmistokehitysprojektit yleensä epäonnistuvat. Projektin aikana saattaa tulla ilmi riskejä, joihin ei ole osattu varautua, jonkin riskin todennäköisyys, vaikutus tai vakavuus saattaa muuttua, joten riskien hallintaa on jatkettava koko projektin ajan ja projektisuunnitelmaa voidaan joutua muuttamaan.

Kriittisin vaihe ohjelmistokehityksessä on suunnittelu- ja määrittelyvaihe, jossa tehdyt virheet ovat aina vakavia. Ohjelmistoprojektin alkuvaiheessa tehdyt virheet voivat seurata projektia sen loppuun asti ja ne saattavat vaikeuttaa vielä jatkokehitystä ja ylläpitoa.

Epäonnistumiset ovat monisyisiä. Loppuun viedyistä projekteista kaikissa meni jotain pieleen jo valmisteluvaiheessa. Siellä vaikeimpia ovat arkkitehtuuriin, strategiayhteyteen ja prosesseihin liittyvät kysymykset. Rakentamisvaiheessa kaikkinaiset määrittelyt ja toteuttaminen sekä testaaminen ovat pulmapaikkoja. (Tietoviikko 2010).

Ohjelmistotuotannon ongelmat aiheutuvat usein virhearvioinneista ja yllättävistä teki-
jöistä, joihin ei ole osattu varautua suunnitteluvaiheessa. Usein ohjelmistot joudutaan julkaisemaan puolivalmiina, koska niiden kehitykseen on kulunut enemmän aikaa, kun osattiin odottaa. Puuttuvia toiminnallisuuksia toteutetaan usein julkaisun jälkeen, joka voi puolestaan aiheuttaa mm. käyttökatkoja tuotannossa olevaan versioon, palveluiden hidastumista ja joskus jopa tietoturvariskejä.

Yleisiä virheitä ja riskejä, jotka aiheutetaan usein jo suunnitteluvaiheeseen ovat:

- Aikataulutuksen epärealistisuus
- Riittämättömät resurssit
- Integroitavien ohjelmistojen eri versiot
- Teknisten alustojen yhteensopimattomuus
- Väärät odotukset tekniikoille
- Tuotteenriskit (laatu, käytettävyys, suorituskyky, tietoturva)
- Jatkuvat muutokset määrittelyissä ja vaatimuksissa
- Koodimuutokset, tietokantamuutokset
- Riskien huomiotta jättäminen
- Puutteet dokumentoinnissa ja vaatimusten hallinnassa

Jäsenrekisterisovellus toteutettiin vapaaehtoistyönä kehitysprojektina yhdistykselle, joten sen riskit poikkeavat jonkin verran kaupallisista projekteista mm. aikataulun, budjetoinnin ja resursoinnin osalta, jotka yleisesti ottaen ovat projektiohjauksen suurimmat huolenaiheet. Jäsenrekisterisovelluksen suurimmat riskit ovat lähinnä tietoturvariskejä, sillä toteutuessaan ne ovat lähes ainoita, joista voi tulla pahempia seurauk-

sia. Seuraavissa kappaleissa on pohdittu yleisesti projektien riskejä erityisesti ohjelmistotuotannon näkökulmasta.

5.10.1 Aikatauluriskit

Ohjelmistoprojekteihin kuluva aikaa on usein vaikea määritellä etukäteen. On yleistä, että aikataulut ja budjetit ylitetään ja se voi joskus aiheuttaa jopa suurten liiketoimintariskien toteutumisen. Epärealistiset aikataulut ovat ohjelmistotuotannossa edelleenkin arkipäivää ja liian kireät aikataulut aiheuttavat suurimman osan ohjelmistotuotannon riskeistä.

Aikatauluriskeihin voidaan varautua etukäteen mm. priorisoimalla sovellukseen haluttavat toiminnot määrittelyvaiheessa. Näin sekundaarisia toimintoja voidaan tarvittaessa siirtää jatkokehityskohteiksi, jotta tärkeimmät toiminnot saadaan toteutettua aikataulussa. Priorisoinnissa toiminnot on luokiteltava siten, että tiedetään mitkä toiminnot ovat pakollisia ja mitä on tehtävä ensimmäisenä tai mitä ilman voidaan toimia ennen jatkokehitystä. Priorisoinnissa tärkeimmät asiat ovat tietysti tietoturvaan liittyvät toiminnot ja jos sovellus on julkaistava puolivalmiina, halutaan pääsääntöisesti, että se näyttää valmiille loppukäyttäjille päin.

5.10.2 Riittämättömät resurssit

Mitä enemmän kehitettävää ohjelmistoa joudutaan räätälöimään tietyn organisaation tarpeisiin ja mitä enemmän se sisältää integraatioita, sitä useampia eri toimittajia yleensä tarvitaan. Suomessa ohjelmistotalot, jotka tekevät nimenomaan räätälöityjä ohjelmistoja, ovat melko pieniä ja resursseja on vähän. Integroitavat sovellukset ovat usein eri yritysten tuotteita ja rajapintojen koodauksessa on kaikilta toimittajilta saatava resursseja projektiin. Ei siis riitä, että yhdeltä toimittajalta tai alihankkijalta saadaan riittävät resurssit, vaan sovelluskehitykseen liittyy usein useita organisaatioita.

Koska ohjelmistotoimittajat ovat usein pieniä yrityksiä, joissa tietty osaaminen voi olla hyvinkin yhteen henkilöön liittyvää, poissaolot ja lomat aiheuttavat usein viivytyksiä projekteihin. Samalla toimittajalla voi olla käynnissä useita projekteja, joten resurssit on jaettava usealle taholle. Jos projektissa oleva avainosaaja vaihtaa työpaikkaa voi korvaavan osaajan saaminen olla vaikeaa ja jokaisen projektiin uutena tulevan työntekijän perehdytys projektiin vie aikaa.

Toisaalta ohjelmistoprojekteihin ei kannata varata liikaa resursseja, sillä se saattaa monimutkaistaa projektin etenemistä ja vaikeuttaa tiedonkulkua, joka on usein kriittistä myös muissa kun ohjelmistoprojekteissa.

Keskeneräistä ei saa ottaa käyttöön, ei toimintatapaa eikä järjestelmää. Asiat karkaavat käsistä kun ruvetaan oikein räätälöimään, sitä tulee välttää. Toimittaja on otettava hallintaan alusta alkaen. Kaikkinaisia resursseja on hankkeissa aina liian vähän, samoin liiketoiminnan aitoa osallistumista. Projektin valmistelua ja projektikulttuuria yleensä pitää nykyisestä kohentaa. (Tietoviikko 2010).

5.10.3 Integraatiot ja tekniset alustat

Arkkitehtuuri ja muut tekniset alustat ovat aina kriittisiä, sillä niitä on vaikea käydä muuttamaan enää myöhemmin. Integraatiot vaativat yleensä oman suunnitelmansa. Useiden sovellusten integrointi yhteen saattaa aiheuttaa suuriakin muutoksia aikatauluun, sillä kaikkien integroitavien sovellusten alustat ja arkkitehtuurit eivät aina välttämättä sovi suoraan yhteen. Jos valmiita rajapintoja eri alustoille ei ole valmiina, ne on tehtävä erikseen, mikä puolestaan venyttää aikataulua. Mitä enemmän integroitavia sovelluksia on, sitä monimutkaisempaa niiden integrointikin on. Jos integraatiot on suunniteltu hyvin, myös jatkossa uusien ohjelmistojen tai toimintojen sulautus järjestelmään on helpompaa, koska hyvin suunniteltua ja dokumentoitua alustaa on helpompi käyttää.

Jos integroitavien sovellusten määrä on suuri (kuten erityisesti useissa sähköiseen asiointiin liittyvissä projekteissa), kannattaa joskus miettiä olemassa olevien arkkitehtuurikerrosten toimivuutta ja mahdollisesti myös niiden uusimista, jotta tulevaisuudessa uusien sovellusten tai niiden osien lisääminen olisi helpompaa. Hyvin suunniteltu arkkitehtuuri ja integrointi nopeuttavat usein myös ohjelmistojen toimintaa. Esimerkiksi, jos kaikki sovellukset saadaan käyttämään samaa rajapintakerrosta, tiedot voidaan hakea kerran ja kaikki sovellukset voivat käyttää tietoa. Eri sovellusten ei tarvitse hakea tietoja erikseen ja tiedon hakuun eri tietokannoista kuluu vähemmän aikaa, mikä puolestaan nopeuttaa ohjelmistojen toimintaa ja parantaa näin myös laadukkuutta. Rajapintojen hyvä dokumentointi ja suunnittelu helpottavat myös uusien versioiden käyttöönottoa ja sulautusta järjestelmään.

Usein suurissa ja kriittisissä ohjelmistoprojekteissa käytetään testialustaa, joka on konfiguraatioltaan tuotantokäytössä olevan sovelluksen ja/tai sovellusalustan mukainen versio. Testialustalla voidaan testata uudet toiminnot, koodimuutokset ja integraatiot ilman, että tuotannossa olevan version käyttö häiriintyy. Näin saadaan myös uuden version asennuksesta aiheutuvien mahdollisten käyttökatkosten aikaa vähennettyä.

5.10.4 Muutoksen hallinnan riskit

Yksi yleisimmin toteutuvista riskeistä ohjelmistoprojekteissa on jatkuvat muutokset vaatimusmäärittelyssä. Muutoksia tulee yleensä siksi, että sovellusta ei ole suunniteltu riittävästi etukäteen ja myöhemmin huomataan, ettei suunnitelmassa ole otettu huomioon tärkeitä toimintoja tai sovellukseen halutaan koko ajan jotain lisää.

Vaatimusmäärittely toimii usein myös projektiosapuolten välillä sopimuksena siitä, mitä projektissa on tarkoitus toteuttaa. Jos määrittely muuttuu jatkuvasti, siitä voi aiheutua ongelmia myös budjetin osalta, sillä toteutettava sovellus voi kasvaa huomattavasti suuremmaksi tai monimutkaisemmaksi, kun mitä alun perin oli tarkoitettu. Vaatimusten muuttuminen aiheuttaa aina lisätöitä. Pohjonen (2007: 81) kehottaakin esitutkimusvaiheen vaatimusmäärittelyn laajentamista koko ohjelmiston elinkaaren kattavaksi vaatimusten hallinnaksi.

Vaatimusmäärittelyn muutokset aiheuttavat muutoksia usein myös ohjelmakoodiin, mikäli muutokset tehdään ohjelmoinnin aloittamisen jälkeen. Muutokset ohjelmakoodiin, etenkin mitä myöhäisemmässä vaiheessa niitä tehdään, saattavat aiheuttaa vakavia ongelmia. Pienikin koodimuutos voi aiheuttaa muutoksia muiden, jo valmiiden ja testattujen, ominaisuuksien toimintaan ja siksi koodin kommentointi ja riittävä dokumentointi on äärimmäisen tärkeää. Pohjonen valottaa ongelmaa tarkemmin tyypillisellä esimerkillä:

Kehityshankkeen aikana asiakastaho ilmoittaa haluavansa järjestelmään uuden ominaisuuden tai muutoksen. Mikäli tällaisia tilanteita varten ei ole etukäteen sovittua käytäntöä, johtaa tilanne liian helposti siihen, että uusi vaatimus sen tarkemmin analysoimatta sisällytetään kehityshankkeen tavoitteiden joukkoon, mikä puolestaan suurella todennäköisyydellä johtaa budjetin ja aikataulun ylittymiseen (ellei hankkeen suunnitelmiin sitten ole varta vasten jätetty tai muuten vain sattunut jäämään

”löysiä”). Jokaisen hankkeen tulisikin määritellä muutospyyntö-käytäntö (change request), josta käy ilmi, miten uudet vaatimukset käsitellään (Pohjonen 2007: 81).

Aikataulut ja resurssit ovat usein tiukkoja, eikä niissä yleensä ole joustovaraa, eikä sitä kannata etukäteen varatakaan. Tästä syystä muutokset tai lisäykset jätetään usein jatkokehityskohteiksi, mikäli ne eivät ole sovelluksen toiminnan kannalta kriittisiä tai pieniä muutoksia, jotka on helppo toteuttaa.

Jos tietokantojen rakennetta tai sisältöä joudutaan muuttamaan kesken kaiken, saate-taan myös valmista koodia joutua muuttamaan paljonkin. Siksi tietokannat kannattaa suunnitella tarkkaan ja ohjelmoinnissa kannattaa käyttää mahdollisimman paljon funktioita, jotka ovat kaikkien toimintojen käytössä. Näin riittää, että mahdolliset muutokset tehdään vain tiettyihin funktioihin, eikä kaikkien toimintojen koodia tarvitse muuttaa.

5.10.5 Riskit puutteellisesta dokumentoinnista

Puutteellinen dokumentointi on yksi suurimmista ongelmista ohjelmistotuotantoon liittyen. Riittävä dokumentointi on tärkeää, että kaikki projektiin osallistuvat tietävät ja ovat yksimielisiä siitä mitä ollaan tekemässä. Vaatimusmäärittelyn riittävä yksityiskohtaisuus korostuu erityisesti silloin, kun ohjelmistoprojektissa on mukana useita toimittajia ja alihankkijoita. Puutteellinen dokumentointi aiheuttaa ongelmia myös myöhemmin ohjelmiston elinkaareissa, sillä se vaikeuttaa ylläpitoa ja päivitystä.

Pohjosen (2007: 79) mukaan ongelma voi olla myös päinvastainen eli projektin byrokratia nousee hankkeen tavoitetta tärkeämmäksi, eikä projektiorganisaatio pääsen eteenmään hankkeessa. Pohjonen toteaa seuraavaa:

Todellinen tarvittava dokumentaation määrä ja laatu riippuu kehityshankkeen koostaja monimutkaisuudesta. Pienne projektin tiedonkulku ja dokumentointi saattavat hoitua vähäiselläkin dokumentaatiolla (Pohjonen 2007: 81).

5.10.6 Riskien välttäminen jatkokehityksessä

Ohjelmistonprojektin suunnitteluvaiheessa on syytä pohtia toteutusta myös tulevaisuutta silmällä pitäen. Ohjelmistoja ei tietenkään kannata toteuttaa jo valmiiksi vanhoilla tekniikoilla tai alustoille, joiden tiedetään vaihtuvan heti projektin valmistuttua.

Määrittelyvaiheessa tulee usein myös esiin toimintoja, joita ei toteuteta projektin puitteissa. Nämä toiminnot ovat kuitenkin hyvä dokumentoida vaatimusmäärittelyyn myöhempää kehitystä varten. Ohjelmiston suunnittelu voidaan siis helpommin toteuttaa niin, että sen jatkokehitys on huomioitu jo etukäteen. Näin toteutuksessa voidaan huomioida myös mahdolliset tulevat ominaisuudet jotka on helpompi lisätä myöhemmin, kun ne on huomioitu jo aiemmin.

6 TOTEUTUSVAIHE

Toteutusvaihe sisälsi itse sovelluksen toiminnallisuuden ohjelmoinnin ja tietokannan luonnin. Projektin osituksesta on kerrottu aiemmissa kappaleissa (kts. 3 PROJEKTIN OSITUS JA VAIHEISTUS s. 14), mutta toteutusvaiheessa jäsenrekisteriprojekti jakautui automaattisesti yksityiskohtaisempiin osiin työläjin ja järjestelmän toimintojen perusteella.

Jäsenrekisterin määrittely- ja suunnitteluvaiheessa sovellus jakautui kahteen käyttäjäkuntaan: jäseniin ja jäsenrekisterin ylläpitäjiin. Jäsenten ja rekisterin ylläpitäjien käyttöliittymistä löytyvät osittain samat toiminnot tietyin rajoituksin. Jäsenrekisteriin suunnitellut toiminnot jakautuivat myös erillisiin osiin, kuten käyttäjänhallinta, jäsentietojen hallinta, laskutus ja jäsenten välinen kommunikointi. Sovellus toteutettiin osittain näitä osituksia silmällä pitäen, sillä niistä löytyi paljon yhteneväisyyksiä. Toisaalta jo suunnitteluvaiheessa havaittiin tiettyjä yhteneväisyyksiä sovelluksen toimintojen välillä ja toteutus tehtiin osittain sen mukaan, mitä funktioita voidaan uudelleen käyttää jossakin toisessa osiossa.

6.1 Tietokannan luonti ja migratointi

Jäsenrekisterisovelluksen toteutusvaihe aloitettiin toteuttamalla aiemmin suunniteltu tietokanta. Tietokantaan tarkoitetut tiedot olivat jo osittain olemassa, joten niitä oli

tarkoitus käyttää alusta asti myös testitietoina, joiden avulla nähtäisiin helpommin järjestelmän toimivuus oikeassa käytössä.

Kotkan Melojat ry:n vanhan Excel-pohjaisen jäsenrekisterin tiedot oli kirjoitettu muotoon, joka ei ollut yhteneväinen uuden jäsenrekisterin tietokannan kanssa. Muun muassa osoitetiedot oli kirjoitettu yhteen Excel-taulukon soluun, kun taas tietokannassa tiedot ovat pilkottuina niiden pienimpään yksikköön normalisointisääntöjen (kts. 5.7 Tietokannan suunnittelu ja normalisointi s. 67) mukaisesti siten, että esimerkiksi jokaiselle osoitteen osalle (katuosoite, postinumero ja postitoimipaikka) on oma kenttensä. Tietojen migratointi aloitettiin luomalla Exceliin VBA-makroja, joiden avulla mm. osoitetiedot eroteltiin omiin soluihinsa. Tietokantatietojen migratoinnilla tarkoitetaan tässä siis olemassa olevien tietojen saattamista siihen tilaan, että ne voidaan siirtää uuteen tietokantaan ja että ne ovat yhtenäisiä uuden tietokantarakenteen kanssa. Kun Excel-taulukon muoto vastasi uuden jäsenrekisterin tietokannan muotoa, se tallennettiin CSV-formaattiin, jonka avulla tiedot oli helppo viedä suoraan uuteen tietokantaan.

6.2 Ohjelmointi ja ohjelmoinnin aikainen testaus

Ohjelmoinnin toteutus oli melko suoraviivainen toimenpide. Ensin tehtiin valmiiksi sovelluksen kannalta tärkeimmät toiminnallisuudet, joita voitiin hyväksikäyttää muiden toimintojen kehityksessä. Ohjelmointivaihe aloitettiin käyttäjähallinnan toteutuksella. Luotiin kaksi käyttäjätasoa, joille määriteltiin erilaiset oikeudet tulevan järjestelmän toimintoihin. Käyttäjänhallintaan toteutettiin käyttäjätunnusten luonti, muokaus ja poisto -toiminnot sekä salasanan vaihto.

Koska jäsenrekisterin perustoiminnot kuten jäsentietojen käsittely olivat yhteneväiset molempien käyttäjätasojen välillä, ideana oli tehdä yksi ja sama käyttöliittymä molemmille käyttäjätasojen siten, että käyttöliittymän toimintoja ja näkymiä rajoitetaan käyttäjätason mukaisesti. Perustoiminnot sisälsivät myös funktioita, joita tarvittiin muiden toimintojen toteutuksessa, joten ne ohjelmoitiin ensimmäisinä. Uusi jäsenhakemus tehtiin myös samalla pohjalla, kuin jäsentietojen lisäys.

Ohjelmoinnissa pyrittiin kiinnittämään huomiota myös ohjelmointityyliin ja koodin rakenteisiin, jotta sen korjaaminen ja muokaus myöhemmin olisi helpompaa. Koodin luettavuuden parantamiseksi pyrittiin käyttämään yksikertaisia rakenteita ja käyttä-

mään kuvaavia nimeämiskäytäntöjä. Suurena osana sovelluksen ohjelmointia on koodin kommentointi.

Jäsenrekisterin toimintoja testattiin koko ajan ohjelmoinnin yhteydessä ja jokaisen osion valmistuttua, niille tehtiin testaus, jossa testattiin osion toimivuutta sekä sen reagointia virheellisiin syötteisiin ja sovelluksen antamia virheilmoituksia (esim. liian lyhyt salasana, sähköpostiosoitteen oikeellisuus jne.) (vrt. 5.9 Testaussuunnitelma s. 72).

6.3 Tilisiirtolomakkeen toteutus

Tilisiirtolomake on ainoa sovelluksesta tulostettava dokumentti, jonka on noudatettava tarkkaa ohjeistusta. Finanssialan Keskusliiton määrittelemän ohjeen mukainen tilisiirtolomake toteutettiin Kotkan Melojat ry:n jäsenrekisteriin erillisen CSS-tiedoston avulla.

Standardin mukaisen tilisiirtolomakkeen ohjelmointi oli yksi työläimpiä tehtäviä toteutuksessa erittäin yksityiskohtaisten sääntöjen vuoksi. Ohjeistuksessa on määritelty tarkalleen jokaisen tilisiirtolomakkeen kentän koko millimetrin kymmenesosan tarkkuudella, viivanpaksuudet ja fonttien ominaisuudet, joten sen toteutus vaati paljon työtä ja tarkkuutta. Lisäksi laskuun tulostettavan viitenumeron laskevan algoritmin suunnittelu ja toteutus oli työlästä, vaikkakin sen pohjana käytettiin osittain valmista algoritmia.

Laskukirjeet tilisiirtolomakkeineen voidaan tulostaa suoraan sovelluksesta. Laskun kirjeosaan tehtiin ikkunalliseen kirjekuoreen sopiva osoitekenttä. Lasku koostuu laskun erittelystä sekä vapaamuotoisesta tekstistä, jossa on käytetty Kotkan Melojat ry:n hinnastoa. Sovellus hakee laskun tiedot automaattisesti määriteltyjen valintojen mukaan.

7 TESTAUSVAIHE

Ohjelmistotestausta suoritettiin jatkuvasti yhtäaikaaisesti ohjelmoinnin kanssa lasilaatikkotestauksen menetelmiä hyväksikäyttäen (kts. 5.9.1 Lasilaatikkotestaus s. 74). Sovelluksesta testattiin sen toiminnallisuutta, käytettävyyttä, yhteneväisyyttä, syötteitä ja

virheilmoituksia. Sovellusta testattiin eri selaimissa ja lopuksi sille suoritettiin betatestaus.

7.1 Jäsenrekisterisovelluksen funktionaalinen testaus ja betatestaus

Jäsenrekisterisovelluksen testaus muodostui sikäli ongelmalliseksi, että yleensä ohjelmoija ei voi testata omaa koodiaan, sillä sen sisältämille virheille ja puutteille tul-
laan helposti sokeiksi. Tästä syystä ohjelmoijat eivät testaa omaa koodiaan. Tutkimus-
ten mukaan ulkopuolinen testaaja löytää noin 60% koodivirheistä. Koska projektissa
ei ollut saatavilla resursseja funktioiden testaukseen, testausta eri toiminnoille suori-
tettiin, vasta kun koodin kirjoittamisesta oli kulunut jonkin verran aikaa. Sovelluksen
syötteitä testattiin funktionaalisten testausmenetelmien (kts. 5.9.2 Funktionaalinen tes-
taus eli mustalaatikkotestaus s. 74) mukaisesti ja toiminnot testattiin vaatimusmääritte-
lyä vastaan.

Kotkan Melojat ry:n hallituksen jäsenet testasivat jonkin verran sovelluksen toiminto-
ja sen ohjelmoinnin aikana ja jäsenille tarkoitettujen toimintojen valmistuttua niitä tes-
tattiin betatestauksen avulla jäsenistä valitun ryhmän toimesta (kts. 5.9 Testaussuunni-
telma s. 72).

7.2 Html-koodin validointi ja selaintestaus

Validilla html- tai CSS-koodi tarkoitetaan sitä, että web-sivun taustalla oleva koodi on
kirjoitettu sääntöjen mukaisesti. Jos koodi ei ole validia, web-sivun sisältö saattaa nä-
kyä selaimessa väärin, eivätkä hakukoneet osaa välttämättä tulkita sivun indeksointia
oikein. Html- ja CSS-koodin oikeellisuus voidaan tarkistaa erillisellä validaattorilla,
joita on saatavissa ilmaiseksi.

Kotkan Melojien jäsenrekisterin koodin validius tarkastettiin W3C-yhteisön sivuilla
olevalla validaattorilla osoitteessa <http://validator.w3.org/>. Validaattorille annetaan si-
vun URL-osoite ja validaattori tarkastaa koodin oikeellisuuden. Mikäli koodi ei nou-
data standardeja, validaattori antaa selvityksen missä kohtaa koodi poikkeaa säännöis-
tä.

Kotkan Melojien jäsenrekisterin toimivuus testattiin seuraavilla selaimilla:

- Firefox, versio 16
- Internet Explorer, versiot 8 ja 9
- Google Chrome, versio 22
- Safari (testaus iPad:lla)

8 KÄYTTÖÖNOTTOVAIHE

Jäsenrekisterin toiminnot otettiin käyttöön vaiheittain. Betatestauksen yhteydessä käyttäjätunnukset luotiin kolmelle hallituksen jäsenelle ja seitsemälle jäsenelle. Jäsenrekisterin käyttäjäkuntaa on tarkoitus laajentaa asteittain ja luoda tunnuksia tarpeen mukaan halukkaille. Tarkoituksena on tiedottaa jäsenistölle uudesta palvelusta yhdistyksen vajalla olevalla ilmoitustaululla, kotisivuilla ja seuran lehdessä, jonka julkaisu on päätetty aloittaa uudelleen, sekä alkeiskurssien aluksi tuleville jäsenille.

Uudet jäsenet saavat omat tunnuksensa täytettyään sähköisen jäsenhakemuksen ja oikeudet jäsenrekisterin toimintoihin avataan, kun jäsenhakemus on virallisesti hyväksytty yhdistyksen hallituksen toimesta.

9 YLLÄPITO JA JATKOKEHITYS

Jäsenrekisterin jatkokehitystä ja ylläpitoa toteutetaan edelleen. Rekisterin sisältämiä tietoja päivitetään jäsenten ja hallituksen toimesta. Tarvittaessa palveluun lisätään uusia toimintoja tai parannetaan jo olemassa olevia. Jatkokehitystä varten on tehty erillinen testitietokanta, joka ei ole tietosisällöltään yhteneväinen tuotantokäytössä olevan tietokannan kanssa, mutta jonka avulla uusia toimintoja voidaan testata ennen niiden julkaisua. Näin vältetään käyttökatkoilta ja siltä, etteivät testaamattomat tai virheelliset koodit pääse sekoittamaan käytössä olevaa tietokantaa tai vaikeuttamaan sovelluksen käyttöä.

9.1 Varmuuskopiot

Tietokannan tiedoista otetaan ajoittain varmuuskopio siltä varalta, että palvelimella olevan tietokannan tiedoille tapahtuisi jotain ei-toivottua. Varmuuskopioiden ansiosta tietokanta voidaan palauttaa johonkin aiempaan, toimivaan tilanteeseen. Sovelluksen tiedostoista on olemassa kopiot ja kuten edellä on mainittu, uusia tai paranneltuja toi-

mintoja testataan käytössä olevan sovelluksen ulkopuolella, jotta testaus ei häiritse palvelun käyttöä.

9.2 Jatkokehityskohteita

Jo suunnitteluvaiheessa pohdittiin joitakin jatkokehityskohteita, joiden toteutusta ei katsottu kiireelliseksi tai oleelliseksi toteuttaa heti. Alla olevassa listassa on lueteltu joitakin mahdollisia jatkokehityskohteita jäsenrekisterisovellukselle.

- Virtuaaliviivakoodin lisäys laskuun
- Sähköpostilasku
- Tiedotteiden ja tiedostojen lisäys palveluun
 - o Yhdistyksen hallituksen jäsenille tulee sähköpostitse tiedotteita muilta seuroilta ym. Tällaisia tiedotteita voitaisiin antaa tiedoksi yhdistyksen jäsenille jäsenrekisterisovelluksessa.
- Kaluston vikailmoitukset
 - o Yhdistyksellä on kalustonhoitaja, jonka tehtäviin kuuluu korjata ja huoltaa yhdistyksen kajakkeja. Kalustonhoitaja saa tiedon jonkin kajakin korjaustarpeesta ainoastaan suullisesti yhdistyksen vajalla, eikä vioista muisteta aina ilmoittaa. Uuteen sovellukseen voitaisiin lisätä toiminto, jossa jäsenet voisivat tehdä vikailmoituksia kalustonhoitajalle.
- Retkille ilmoittautuminen
 - o Kotkan Melojat ry järjestää melontaretkiä, joille voitaisiin ilmoittautua järjestelmän kautta.
- Kalustovaraukset
 - o Kaluston käyttö on vapaata niille seuran jäsenille, jotka ovat maksaneet kalustonkäyttömaksun. Kalustoa ei kuitenkaan saa pitää käytössä yli vuorokautta ilman erillistä lupaa. Uuteen järjestelmään voitaisiin lisätä toiminto, jonka avulla voitaisiin tehdä kalustovaroja jollekin tietylle päivälle tai useamman päivän kestävää käyttöä varten kuten Suomi Meloo -tapahtumaa varten.
- Vajan säilytyspaikkojen hallinta
 - o Yhdistyksen vajoilla on säilytyspaikkoja jäsenten omien kajakkien säilytykseen säilytysmaksua vastaan. Säilytyspaikat ovat vähissä, eikä kaikissa vajoilla olevissa kajakeissa ole merkintää kenelle ne kuuluvat. Kaikista säilytyksessä olevista kajakeista ei myöskään makseta säilytysmaksua. Jäsenrekis-

terisovellukseen voitaisiin tehdä toiminto, jonka avulla kajakkien säilytystä yhdistyksen tiloissa voitaisiin valvoa ja laskuttaa.

9.3 Tietoturvan uudelleenarviointi

OECD (Organisation for Economic Cooperation and Development) antoi tietoturvasuusperiaatteita koskevat suositukset vuonna 1992 ja niitä uudistettiin vuonna 2002. Tietoturvasuusperiaatteita on yhdeksän ja viimeinen niistä edellyttää tietoturvan jatkuvaa uudelleenarviointia.

Tahojen tulisi arvioida uudelleen tietoturvasuutta ja tarpeen mukaan tarkistaa sitä koskevat politiikkansa, käytäntönsä, toimensa ja menettelynsä. Uusia ja muuttuvia uhkia ja haavoittuvuuden lähteitä löydetään jatkuvasti. Niihin vastatakseen tahojen tulee jatkuvasti arvioida uudelleen ja tarkistaa kaikki turvasuuteen liittyvät tekijät. (Valtiovarainministeriö 2002: 14).

Jäsenrekisterin tietoturvaa arvioidaan jatkuvasti tietoturvasuusperiaatteiden mukaisesti.

10 MUITA NÄKÖKULMIA

Ohjelmistoprojekteilla ei yleensä katsota olevan suuriakaan ympäristövaikutuksia, mutta koska melonta on luonnonläheinen laji, voidaan jäsenrekisterisovelluksen kaltaisen foorumin katsoa vaikuttavan välillisesti myös ympäristöasioihin. Kanoottiliiton (nyk. Suomen Melonta- ja soutuliitto) ympäristöohje, joka on julkaistu osana melojan ympäristöohjelmaa toteaa seuraavaa:

Ympäristön kannalta lajin lähtökohdat ovat loistavat: liikutaan omin lihasvoimin ja luonnonvesissä, olipa kohteena sitten retkisaari, paikallinen uimaranta, hikilenkki tai kilpailun maali. Kuten urheiluaatteeseen kuuluu, aina on mahdollista kehittyä ja päästä korkeammalle, pidemmälle, nopeammin. Melonnan ympäristövaikutusten tarkastelussa ja toiminnan kehittämisessä entistä laadukkaammaksi on kyse samasta asiasta. Panostamalla ympäristöasioihin me voimme olla vieläkin parempia (Suomen Melonta- ja soutuliitto 2000: 2).

Vaikka projektin toteutuksella ei siis sinänsä ollut suurempia ympäristövaikutuksia, sovelluksen käyttöönotto ja toimintojen optimointi vähentää paperin- ja musteenkulutusta ja vähentää näin samalla syntyvän jätteen määrää. Jatkokehityskohteissa mainittu sähköpostilaskutuksen toteuttaminen (kts. 9.2 Jatkokehityskohteita s. 87), vähentäisi kulutusta entisestään. Lisäksi jatkokehityskohteissa on mainittu tiedotteiden ja tiedostojen lisäysmahdollisuus palveluun. Tällaisen toiminnon avulla voitaisiin samalla lisätä jäsenistön ympäristötietoutta jakamalla ympäristöoppaita, tietoa jokamiehen oikeuksista ym.

11 JOHTOPÄÄTÖKSET JA YHTEENVETO

Vaikka ohjelmistoprojektien sudenkuopat ovat tarkalleen tiedossa ja niitä osattiin odottaa, sorruttiin niihin jossain määrin tässäkin projektissa. Projektin alkuperäinen dokumentointi lähinnä vaatimusmäärittelyn osalta oli lähinnä kokoelma suurpiirteisiä näyttöjen rautalankakuvia ja yleisiä kuvauksia toiminnoista. Dokumentit tarkentuivat kuitenkin koko ajan, sillä projektin vaiheita suoritettiin osittain päällekkäin ja yhden osion tarkemman suunnittelun ja toteutuksen jälkeen saatiin paljon käyttökelpoista tietoa muiden osioiden tekoon. Jo alkuvaiheessa huomattiin suunnittelun ja dokumentoinnin merkitys, sillä tarkat kuvaukset käyttöliittymistä ja toiminnoista helpottivat erittäin paljon sovelluksen toteutusta. Toisaalta, koska suunnittelu ja toteutus tehtiin saman henkilön toimesta, toteutusvaiheessa voitiin korjailla joitakin toimintoja tai yksinkertaistaa niitä.

Koska jäsenrekisterisovellus toteutettiin vapaaehtoistyönä oman työajan ulkopuolella, ei sen toteutus ollut täsmällistä ja jatkuvaa, vaan sitä tehtiin henkilökohtaisen aikataulun ja resurssien sallimissa puitteissa.

Sovellus otettiin käyttöön keskeneräisenä (betatestaus), kun jäsenrekisterin päätoiminnot oli saatu kuntoon ja kun jäsenille tarkoitettu osio oli valmis. Sovellukseen kuuluvan laskutusosion ohjelmointia jatkettiin vielä myöhemmin, sillä laskutus tehdään pääsääntöisesti vain kerran vuodessa, eikä sen valmistumisella katsottu olevan kiirettä. Toisaalta jäsenrekisterin ylläpito ja laskutus kuuluvat seuran sihteerille eli tämän päätöstyön tekijälle, joten puutteet hankaloittavat ainoastaan sovelluskehittäjän omaa työtä.

Resurssien puutteen vuoksi erityisesti testaus aiheutti päänvaivaa, sillä kuten aiemmin on mainittu, oman koodin testaaminen ei ole järkevää. Toisaalta vaikka olen sovelluksen tekijä, olen myös osana kumpaakin sidosryhmää (jäsen ja hallituksen jäsen), joten vaatimukset, ongelmat ja käyttötarkoitukset olivat hyvin tiedossa, eikä erillistä perehtymistä olemassa oleviin toimintamalleihin juurikaan tarvittu. Koska en ole ollut yhdistyksen hallituksessa pitkään, ei objektiivisen näkökulman omaksuminen uusia toimintamalleja suunniteltaessa ollut mielestäni vaikeaa. Osana sovelluksen käyttäjäkuntaa huomasin myös itse monta uutta kehityskohdetta ja sovelluksen käyttöä helpottavia muutoskohteita, joista osa on jo toteutettu ja osa siirretty jatkokehitykseen.

LÄHTEET:

Arkistolaitos. Hyvään tiedonhallintatapaan -esite 2004. Saatavissa:
<http://www.arkisto.fi/uploads/Palvelut/Julkaisut/hyvatiedonhallinta2.pdf> [viitattu 29.7.2012].

Federal Trade Commission (FTC) 2012.
<http://www.ftc.gov/opa/2012/02/2011complaints.shtm> [viitattu: 13.7.2012].

Financial Risk Management News And Analysis 2012. <http://www.risk.net> [viitattu 13.7.2012].

Finanssialan Keskusliitto. Yhtenäisen euromaksualueen toteutuminen suomessa: Suomen kansallinen SEPA –siirtymäsuunnitelma. Versio 5 2012.
http://www.fkl.fi/teemasivut/sepa/.../SEPA_siirtymasuunnitelma_v5.pdf [viitattu 16.4.2012].

Finanssialan Keskusliitto. Tilisiirto-opas 2011.
http://www.fkl.fi/materiaalipankki/opaat/Dokumentit/Tilisiirto-opas_2011.pdf [viitattu: 16.4.2012].

Finanssialan Keskusliitto. Kotimaisen viitenumeron rakenne 2009.
www.fkl.fi/teemasivut/sepa/.../kotimaisen_viiiteen_rakenneohje.pdf [viitattu 16.4.2012].

Finlex. 1999. Henkilötietolaki 22.4.1999/523.
<http://www.finlex.fi/fi/laki/ajantasa/1999/19990523> [viitattu 30.8.2012].

Finlex. 1989. Yhdistyslaki 26.5.1989/503.
<http://www.finlex.fi/fi/laki/ajantasa/1989/19890503> [viitattu 30.8.2012].

Haikala I. & Märijärvi J. 1997. Ohjelmistotuotanto. Espoo: Suomen ATK-kustannus Oy.

Järvinen P. 2002. Tietoturva & yksityisyys. Jyväskylä: Docendo Finland Oy.

Jääskeläinen O. MikroPC -lehden artikkeli 2011.

www.mikropc.net/kaikki_uutiset/tietovuoto+14+600+salasanan+lista+julki++joukossa+on+oikeita+salasanoja/a721508 [viitattu 11.10.2012].

Jääskeläinen O. MikroPC -lehden artikkeli 2011.

http://www.mikropc.net/kaikki_uutiset/taas+jattivuoto+500+000+suomalaista+sahkopoistioitetta/a721394 [viitattu 11.10.2012].

Kuluttajavirasto. 2008. Kuluttajaoikeuden linjaus: Maksaminen ja laskutus.

<http://www.kuluttajavirasto.fi/File/4672b687-d8f3-4a93-bd39-9cf0680bb51e/Maksaminen+ja+laskutus.pdf> [viitattu 13.7.2012].

McConnell S. 2002. Ohjelmistotuotannon hallinta. Helsinki: Edita Prima Oy.

Microsoft. Tietokannan normalisoinnin perusteiden kuvaus.

<http://support.microsoft.com/kb/283878/fi> [viitattu 19.10.2012].

Nielsen J. Top 10 Mistakes in Web Design 2011.

<http://www.useit.com/alertbox/9605.html> [viitattu 11.10.2012].

Nielsen J. Ten Usability Heuristics 1990.

http://www.useit.com/papers/heuristic/heuristic_list.html [viitattu 11.10.2012].

Pelin, R. 2011. Projektihallinnan käsikirja. 7. uudistettu painos. Keuruu: Projektijohtaminen Oy.

Pohjonen R. 2007. Tietojärjestelmien kehittäminen. 3. painos. Jyväskylä: Docendo Finland Oy.

Pöysti T. Still V. 1999. Insinöörin etiikka tietoyhteiskunnassa. Helsinki: Insinööriliitto.

Rousku K. Tietoturvasarja – Tietoturvapoliittikka. Osa2/6. MikroPC -lehden artikkeli 2003. <http://mikropc.net/nettilehti/pdf/2003200354.pdf> [viitattu 13.7.2012].

Suomen melonta- ja soutuliitto ry. Melojan ympäristöohjelma.

<http://www.melontajasoutuliitto.fi/@Bin/125645/Melojan+ymp%C3%A4rist%C3%B6ohjelma.pdf> 2000.

Suomen Projekti-Instituutti Oy. [Projektijohtamisen sanasto](http://www.projekti-instituutti.fi/sanasto). <http://www.projekti-instituutti.fi/sanasto> 2012.

Suomen Standardisoimisliitto SFS ry. Ergonomian ja käytettävyyden standardit –esite. <http://www.sfs.fi/files/61/ergonomiasfs.pdf> 2011.

Tietosuoja-valtuutetun toimisto. <http://www.tietosuoja.fi/1698.htm> 30.5.2012.

Tietosuoja-valtuutetun toimisto. Henkilömatrikkelit

www.tietosuoja.fi/uploads/tvsw7q2.pdf 2010.

Tietoviikko -lehden artikkeli. Mistä näitä it-mokia tulee? 10.11.2010.

Valtiovarainministeriö. OECD:n suositus. Tietojärjestelmien ja tietoverkkojen turvallisuusperiaatteet: Turvallisuuskulttuurin kehittäminen. Suomenkielinen käännös. 2002.

Valtiovarainministeriö.

http://www.vm.fi/vm/fi/04_julkaisut_ja_asiakirjat/01_julkaisut/04_hallinnon_kehittaminen/4127/name.jsp 2002.

Viestintävirasto. CERT-FI.

<http://www.cert.fi/tietoturvanyt/2012/06/ttn201206061430.html> 6.6.2012.

Viestintävirasto. CERT-FI.

<http://www.cert.fi/tietoturvanyt/2011/11/ttn201111051616.html>. 5.11.2011.

World Wide Web Consortium (W3C).

http://www.w3schools.com/browsers/browsers_stats.asp 22.10.2012.